AFRL-RI-RS-TR-2008-90
Final Technical Report
March 2008

THE MARYLAND LARGE-SCALE INTEGRATED NEUROCOGNITIVE ARCHITECTURE

University of Maryland

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. V029

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-90 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
THOMAS E. RENZ Work Unit Manager	JAMES A. COLLINS, Deputy Chief Advanced Computing Division Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAR 2008	Final	Sep 05 – Aug 07

4. TITLE AND SUBTITLE

THE MARYLAND LARGE-SCALE INTEGRATED NEUROCOGNITIVE ARCHITECTURE

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-05-2-0272

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)

J. Reggia, M. Tagamets, J. Contreras-Vidal, D. Jacobs, S. Weems, W. Naqvi, C. Yang

5d. PROJECT NUMBER
BICA

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
04

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland
University College
1103 Lee Building
College Park MD 20742-5151

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RITC
525 Brooks Rd
Rome NY 13441-4505

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-90

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-0941

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Recent progress in neural computation, high performance computing, neuroscience and cognitive science suggests that an effort to produce a general-purpose, adaptive machine intelligence is likely to yield a qualitatively more powerful system than those currently existing. Here we outline our progress in developing a framework for creating such a large-scale machine intelligence, or *neurocognitive architecture* that is based on the modularity, dynamics and plasticity of the human brain. We successfully implemented three intermediate-scale parts of such a system, and these are described. Based on this experience, we concluded that for the short term, optimal results would be obtained by using a hybrid design including neural, symbolic AI, and artificial life methods. We propose a three-tiered architecture that integrates these different methods, and describe a prototype "mini-Roboscout" that we implemented and evaluated based on this architecture. We also examined, via computational experiments, the effectiveness of genetic programming as a design tool for recurrent neural networks, and the speed-up obtained for adaptive neural networks when they are executed on a graphical processing unit. We conclude that the implementation of a large-scale neurocognitive architecture is feasible, and outline a roadmap for proceeding.

15. SUBJECT TERMS
Machine Intelligence, Neurocognitive Architecture, Hybrid AI, Recurrent Neural Network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thomas E. Renz
U	U	U	UL	50	19b. TELEPHONE NUMBER (*Include area code*) N/A

Table of Contents

List of Figures

List of Tables

Introduction

The idea of creating a general purpose machine intelligence that captures many of the features of human cognition goes back at least to the earliest days of artificial intelligence (AI) and neural computation. In spite of more than a half-century of research on this issue, there is currently no existing approach to machine intelligence that comes close to providing a powerful, general purpose human-level intelligence. For example, while general cognitive architectures [Rosenblum et al, 1993; Anderson et al, 2004] have been studied for many years and have been used to model many specific aspects of human behavior, they have been less successful in scaling up to real world applications, and are limited by being rooted in rule-based (production system) processing. There have also been fairly general AI models of knowledge representation and inference, such as those based on first-order predicate calculus and state space search methods [Brachman & Levesque, 2004; Russell & Norvig, 2003; Sowa, 2000]. While these general AI methods are widely applicable, they are sometimes called "weak methods" because they have proven less effective in applications and are computationally expensive. General purpose neural network methods such as backpropagation and self-organized feature maps have also been very successful in specific applications involving learning, such as pattern recognition, data visualization, and autonomous vehicle control, but have not been extended to many aspects of cognition. Many more methods have been studied in cognitive science, AI and neural computation, but the common experience seems clear: success has come in specific, focused domains, and not in the form of a general, human-like ability to solve problems and learn.

In spite of this limited success, we believe that a renewed effort to produce a general purpose and adaptive machine intelligence is timely, likely to yield qualitatively more powerful approaches to machine intelligence than those currently existing, and certain to lead to substantial research progress in cognitive science, AI and neural computation. Our optimism in this regard comes from the convergence of three advances:

- Experiments and discoveries in cognitive science and neuroscience are revealing key aspects of human memory, reasoning and learning mechanisms and their neurobiological basis, e.g., via the use of fMRI and other functional measurements.
- Methods for constructing intermediate-scale modular neural systems have become increasingly effective and refined; the task now is to expand these systems, and to assemble and integrate them in a single framework.
- Progressively more powerful and less expensive computer hardware is becoming available, including non-standard high-performance computing architectures that make possible highly parallel computations.

These advances suggest that progress in creating a powerful, general purpose machine intelligence will come from creating a modular but integrated cognitive architecture that is inspired by human brain organization and supported by a high-performance computing platform.

When one considers the broad range of problems faced by people on a routine basis, it quickly becomes evident that we bring to bear a remarkable range of abilities during problem solving in an *integrated* fashion. Such integration will be essential for a situated, general-purpose machine intelligence to exhibit human competitive (or better) intelligence. In the following, it is

important to recognize that this integration will need to occur along at least two related but largely orthogonal dimensions. The first dimension of integration, *behavioral tasks*, spans the broad range of tasks an intelligent agent must perform, often concurrently. The second dimension of integration, *cognitive mechanisms*, spans the underlying information processing algorithms required to support these individual behaviors/tasks. These include a variety of memory and representation mechanisms, a broad range of reasoning algorithms (deductive inference, causal/explanatory or abductive inference, etc.), methods for generating and/or interpreting temporal sequences of events, learning procedures at multiple levels that lead to improved performance, and top-down control mechanisms that coordinate all of these memory, reasoning, and learning methods. While there are many computational systems today that can produce a reasonable level of performance on one or a few aspects of such behavioral tasks and cognitive mechanisms, no single existing system encompasses the broad array of behaviors and algorithms listed above. Further, it is not enough just to include all of these specific abilities within a single system: they must also act together in an effective and coordinated fashion.

In this context, we believe that the *long-term goal* of creating a general-purpose machine intelligence will best be served by pursuing a computational model that is directly based on the hierarchical and modular organization, dynamics, and plasticity of the human brain, especially the neocortex and its interactions with subcortical structures. Why pursue a neuromorphic/brain-inspired architecture? One reason is that the human brain is currently the only known entity capable of exhibiting robust general intelligence in the form of integrated problem solving, language processing, planning, creative design, and learning. In short, the brain provides the only proof-of-existence that such an integrated intelligent entity is possible, and it is the only known system that encompasses information processing mechanisms sufficient to produce human-level cognition. These mechanisms are based on an underlying neural foundation that inherently supports massively parallel computations, something that is necessary for real-time operation and robustness to damage. Our judgment is that a large-scale computer system modeled after the human cerebral cortex (neocortex), the part of the human brain most closely related to problem solving and cognition, as well as closely integrated non-neocortical brain structures (thalamus, hippocampus, basal ganglia, cerebellum, etc.), is currently the best bet for a truly qualitative advance in machine intelligence over the long term. The following sections of this report present a conceptual framework in which to develop a large-scale neurocognitive architecture of the sort we envision, along with some preliminary results supporting the plausibility of this framework.

While this long-term goal provides a clear target for a successful, general-purpose machine intelligence, it raises the question of what the optimal strategy is for attaining that goal while simultaneously making progress over the short term of the next five years. One strategy would be to immediately commence implementing a large-scale neuromorphic architecture that spans all of cognition. However, our current knowledge of brain function still contains substantial gaps and uncertainties, and our understanding of how to use contemporary neural computation methods effectively to capture some aspects of cognition is also limited. Accordingly, we believe that the *optimal short-term strategy* is to develop a hybrid architecture that combines neurobiologically-inspired methods and cognitively-inspired methods within a unified framework. By “cognitively inspired methods”, we mean more conventional symbolic and numeric methods from cognitive science and AI rather than neural computational methods.

Methods, Assumptions and Procedures

Given that the human brain is the only known system capable of general cognition, it seems prudent to base the design of a general-purpose machine intelligence on the brain's organizational and computational principles, and this is the approach that we take here. Of course, there are widely recognized barriers to such a neurobiologically-inspired methodology, and these have deterred past work in this area. The human brain is highly complex, and we currently have an incomplete understanding of the neurobiological basis of many aspects of human cognition. Those aspects of brain function that we do understand reasonably well seem to be primarily low-level sensorimotor and reflex functions, while higher-level cognitive functions are much less understood. Further, the size and complexity of an artificial large-scale neurocognitive architecture would appear to make its implementation very difficult. We believe that these barriers can largely be overcome. The design of complex systems can be facilitated by modularity, and there is continuing steady progress in understanding the biological basis of cognition, led in part by functional imaging and modern electrophysiological methods. Existing neurocomputational models of individual brain systems show that the technology is there for many of the parts needed for a full-scale system, and the difficult challenge now is how to put those parts together effectively into a large-scale and coordinated whole. Further, contemporary high-performance electronic computing hardware and emerging non-standard computing resources indicate that the needed computational substrate is or will soon be available, and will lead to very efficient implementations by ultimately capturing the natural parallelism of neural computations at the hardware level.

In this and the following sections, we present a computational theory of human cognition that is tightly grounded in the hierarchical and modular structure, dynamics, and plasticity of neocortex and other closely coupled subcortical brain structures. While the inspiration for our approach comes directly from the brain, we are *not* trying to develop a veridical model of the brain. Rather, we are extracting the fundamental organizational and processing principles of the nervous system and applying them to create a neuromorphic machine intelligence. These principles include locality of computation, massively parallel processing, hierarchical and modular structure, decentralized control, and a fundamental role for learning and adaptation. Our theory will subsequently serve as the basis for designing a large-scale integrated model of cognition founded primarily upon neurobiological principles, and this will be described in the later parts of this report. While there are many previous theories/models of brain subsystems, to our knowledge no one has ever created an architecture with the broad scope and integrated coverage of brain and cognitive functions that we are considering here. Our neurobiologically-oriented approach focuses on the critical issue of bridging the gap between neuromorphic systems and cognition.

A. Top-Level Overview

Our neuromorphic theory is based upon an underlying architecture having a network of hierarchically organized modules whose structure and function is directly inspired by human neocortical and subcortical organization and brain relationships to cognition. While there are important gaps in our knowledge [Uttal 2001], a great deal is currently known about the mapping of behavior in general and cognitive functions in particular to human brain regions. We thus summarized the results of our recent efforts to compile a listing of important known function-to-

brain relationships as a separate report [Tinerella et al, 2006]. Cataloging these relationships between cognitive functions and brain regions proved to be an ambitious goal, given the uncertainties and even disagreements about the representation of some aspects of memory, language, and other cognitive functions in the brain. Further, the mapping is not really one-to-one in that some cognitive functions are distributed over multiple brain regions, and some regions contribute to multiple functions [Mesulam, 1990].

The basic conclusion that comes from critically examining current knowledge of human brain structure and function is that the brain's architecture can best be viewed as composed of repeating and nested functional modules. The hierarchical organization is roughly

$$\text{brain} \rightarrow \text{systems} \rightarrow \text{areas/nuclei} \rightarrow \text{local circuits} \rightarrow \text{neurons}.$$

For example, in the neocortex the local circuit modules are cortical columns whose inter-columnar connectivity is extensively (but only partially) documented in the voluminous neuroscientific data that is available. These columns are often viewed as the basic functional units of cortex [Mountcastle, 1998]. At the next level up, modules correspond to cortical areas that are interconnected by various neuroanatomical pathways and tracts. Concrete examples of histologically-distinguishable cortical areas would be the Brodmann areas 1, 2, 3, … which are also labeled in ways related to their functionality (Wernicke's area, prefrontal eye fields, etc.) or anatomical features (supramarginal gyrus, angular gyrus, etc.). These areas can sometimes themselves be divided into subregions, e.g., primary somatosensory cortex region S1 can be viewed as partitioned into hand/arm/trunk regions. Examples of specific pathways/tracts connecting cortical areas are the arcuate fasciculus between Wernicke's and Broca's areas, and callosal connections between corresponding left and right mirror image cortical areas. At the next highest level, interconnected areas are integrated into identifiable functional systems such as the inferior temporal-frontal visual system, the spoken language system, the sensorimotor system, and so forth. Finally, these systems are integrated into a top-level network via the pathways between their components and/or overlapping components. Implicit in this organization are feedforward, feedback, and recurrent connectivity. A similar hierarchical structure can be identified for subcortical regions such as the thalamus and basal ganglia.

In this context, the primary features of our framework for creating a large-scale and general-purpose neurocognitive architecture can be summarized as follows.

- Our architecture is a hierarchical network of nested and iterated modules, inspired by the neurobiological structures outlined above. These modules have spatial relationships to one another, unlike with many neural models, and this has significant implications for connectivity, functionality and learning.

- Functionality in our architecture is provided by the activation dynamics of its modules, occurring simultaneously at multiple levels of the structural hierarchy. In other words, our framework is based on a dynamical systems perspective rather than the primarily logical/symbolic approach used in many mainstream cognitive models in psychology and AI. Cognition is viewed as an emergent property of self-organizing neural processes, not something that is directly "programmed in".

- Both the structural architecture and the neurobiologically-inspired functional mechanisms are guided not only by the need for good performance but also by a drive to

minimize costs (energy use, connectivity, etc.). In part, cost minimization is based upon the strength and nature of functional interactions between brain regions, and is informed by recent human functional imaging data (fMRI) and electrophysiological data (EEG).

- Working memory, executive control functions, and sequential behavioral processing are represented in multiple ways in our theory, including competition between neural modules for activation that influences global control of activity (one aspect of attentional mechanisms), sustained patterns of neural activity in cortical regions, and recurrent connectivity between regions that can gate one another's activity.

- Functions of modules are largely learned, not pre-programmed, so that a module's functionality is determined in part by its location and connectivity, and in part by a "learning agenda" during which different components of the model learn independently in a prescribed, multi-stage fashion before being integrated and trained further collectively, much as occurs in human brain and childhood cognitive development.

- Finally, learning is a continuous process, implying that our architecture can reorganize after damage and partially recover via dynamic reallocation of functionality.

We now turn to making this top-level perspective operational by considering some of the basic design principles in more detail.

B. Structure

Paralleling the hierarchical organization of the human brain summarized above, i.e.,

$$\text{brain} \rightarrow \text{systems} \rightarrow \text{areas/nuclei} \rightarrow \text{local circuits}$$

the structural aspects of our framework are

$$\text{architecture} \rightarrow \text{systems} \rightarrow \text{regions} \rightarrow \text{cells/voxels.}$$

An important emphasis in our approach is that conceptually one is focused on specifying an architecture more at the level of assembling regions into systems and less on specifying low-level details of neurons and their connectivity than in most past neurocomputational work. In other words, while neurocomputational models are often viewed as a "bottom-up" approach to machine intelligence, our conceptual framework takes a "top-down" view of their design.

The lowest level of detail in this framework is the neural *cell* that is loosely intended to model a local volume element, or *voxel,* and its included local neural circuitry, such as a cortical column. The term "cell" here is <u>not</u> related to the concept of a biological cell; it refers instead to a cell of space and its contents in the same way that the term "cell" in computational systems like cellular automata refers to an atomic processing unit. A distinguishing feature of our neuromorphic architecture is that individual neurons within a cell are generally not explicitly represented – the atomic elements used in our model are the cells/voxels and their interconnections. This differs from most neurocomputational models where neurons (or even smaller elements such as dendritic compartments) are explicitly viewed as the atomic units of computation. Our position is that if one wants to develop a large-scale integrated machine intelligence, individual neurons (dendritic trees, molecular structures, etc.) provide too low a level of abstraction at which to start. Some implications of this choice are that the functionality

of local neural circuits must be captured in the internal dynamics of a voxel/cell, and that the dynamics of a cell does not in general match that of an individual neuron. Cells communicate locally in our model via weighted *connections* and have one or more internal activation levels.

Cells in our framework are assembled into regularly structured *regions* that roughly correspond to areas in the cortex, or subcortical neuroanatomic structures such as nuclei in the thalamus or basal ganglia. As illustrated in Figure 1, these cellular arrays or regions have an explicit spatial organization. In the following, we will generally view these arrays as being 2D structures, but there is no reason that other dimensionalities (1D, 3D, etc.) cannot be used, and all that we say below applies equally well in such situations. The regular repetitive cells in arrays provide a simple, uniform base upon which to construct an architecture and define its computational properties, and this uniformity will facilitate a hardware implementation over the long term should that become appropriate. Some implications of an explicit spatial representation are that real-valued distance metrics are relevant, that intra-array connectivity can be an explicit function of geometric (versus topological) distances, and that self-organizing topographic and feature map formation becomes an important functional issue. As illustrated in Figure 1, a region receives inputs and sends outputs to other regions via *pathways*, collections of individual inter-cell connections analogous to identifiable tracts in the central nervous system. Regions also generally have substantial internal recurrent connectivity.

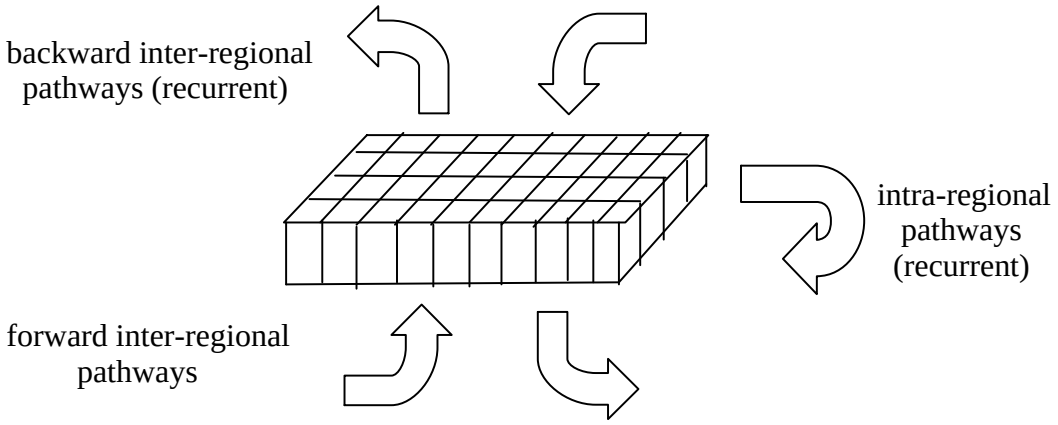
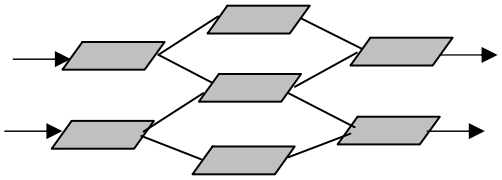


Figure 1. Schematic representation of a generic 2D *region.* Each element is a cell/voxel (volume element) whose functionality captures the dynamics of local neural circuits such as those of a cortical column. Arrows indicate forward (bottom), backward (top) and internal (on the right) connectivity, which is highly recurrent.

A *system* in our framework is the analog of a brain system that is devoted to some class of behavioral function, such as vision, memory, language, etc. As illustrated below in Figure 2, a system is composed of a network of regions that are interconnected via pathways. The explicit spatial organization of regions means that such pathways can be specified as geometrically-meaningful projections or mappings of one region onto another, rather than connection-by-connection. Further, each region like those pictured in Figure 2, viewed as a whole, has one or more associated activation levels distinct from those of its component cells, and each pathway has one or more associated weights distinct from those of its constituent connections. These activation values and weights serve as part of the top-level control mechanism.

Figure 2. A *system* within our architecture is a network of interconnected regions like that in Figure 1, seven of which are shown here. The regions are connected via bidirectional pathways.



Finally, in an analogous fashion, the resultant neurocognitive *architecture* can be viewed as composed of a network of interconnected systems that provide the structural basis of the entire model. Each individual system, viewed as a whole, may have one or more associated activation values, distinct from those of their component regions, and one or more associated weights on their interconnected *couplings* that are distinct from the weights on their inter-regional pathways.

C. Dynamics

At the level of cells, activation dynamics in our model incorporate many features of methods used in contemporary neural networks, and these features are not intended as innovations of this work. Each cell has one or more real-valued activation levels that are repeatedly updated based on incoming activity from other cells in their local neighborhood, or from other regions. Activation rules that govern the updating of a cell's activity are generally expressed as non-linear differential equations, and the behavior of a cell is viewed as a dynamical system having various attractor states. The cells forming a region act collectively, producing region-level attractor states that emerge from the numerous non-linear interactions between activated cells in that region, something that can be viewed as an analog of the "mass action" occurring in the nervous system. Cognitively-relevant information is thus encoded in a region using a distributed representation/encoding (coarse coding). Put otherwise, working memory is represented by sustained activity patterns across regions, where these patterns are the attractor states. Long-term memory is represented in inter-cell connection weight values, or intra-cell parameter values.

In addition to these fairly conventional computational mechanisms, our approach encompasses a number of innovations, or at least non-standard features. One fundamental organizing principle that distinguishes our theory is that neural architectures should be based not only on obtaining good performance, but just as importantly on minimization of costs such as energy use and structural connectivity. Such cost minimization, or parsimony, appears to be an important constraint on brain evolution [Gibbons, 1998], has proven very effective in some of our past work in explaining neocortical dynamics and specialization [Reggia et al, 1992; Shkuro et al 2003], and creates neural architectures that scale up in size better if eventually implemented in hardware. We now describe two ways that this basic parsimony principle is incorporated within our framework.

First, as the cells/voxels that are atomic elements of our model are not neurons, they can exhibit behaviors that are quite different from typical biological neurons in past neural models. For example, a cell in our framework may retain specific details of previously seen input patterns and base its output on such patterns in novel ways. This allows one to capture within a cell's dynamics the functionality of neural circuitry used in some past models of working memory [Tagamets & Horwitz, 1998]. Most relevant here is that a cell/voxel may also exhibit competitive activation dynamics [Reggia et al, 1992] that can substantially reduce intra-regional recurrent connectivity. For example, neocortex has long been recognized to exhibit a Mexican Hat pattern

of activation due to a localized stimulus: a region of evoked activity is surrounded by an annulus of suppressed activity. This is captured in many computational models of cortex by intra-region connections: relatively few short-range excitatory connections and relatively many longer range inhibitory connections. In our model, cells/voxels can competitively distribute their activity, something that is implausible for an individual neuron but is perfectly legitimate for a voxel (neural circuitry) to do. The result is that a Mexican Hat activity pattern is produced without the need for numerous inhibitory connections, greatly simplifying intra-regional circuitry. Distributing neural activity in this competitive fashion implies synaptic connections whose strengths not only change slowly during learning as in most neural models, but also change very rapidly to direct the spread of activity. Such "fast weights" have become increasingly plausible in recent years with the growing evidence that rapid changes in biological synaptic strengths are a common and important computational mechanism in the brain [Abbott & Regehr, 2004].

A second, more cognitively interesting use of competitive dynamics within our framework is at the higher level of regions and their interconnecting pathways (Figure 2). As noted earlier, regions and pathways also have activation levels and weights associated with them that are distinct from those of their components. The higher-level activation values associated with regions can either be derived from the activations of the region's component cells, such as a time-averaged mean activity level, or imposed by other regions or external entities as top-down control information. Similarly, each inter-region pathway has one or more associated weights distinct from the weights on the individual inter-cell connections that compose the pathway. For example, one weight associated with a pathway is its *gain* indicating the magnitude of its inter-regional effects; dynamically adjusting such a gain alters effective network structure. The key idea is that, in integrating regions into systems, and systems into an architecture, these high-level activations/weights allow regions to turn one another on/off, and for one region to "gate" (enable/disable) the flow of activity between other regions. Such gating is believed to occur, for example, between cortical and subcortical brain regions during motor control and during performance of working memory tasks. We view these high-level inter-regional effects as the basis for implementing competitive and cooperative effects between regions, just as they occur between cells within a region, and for parsimoniously distributing activity. In this way, there is a distributed global control of the flow of activity throughout the overall architecture, and this control process forms one aspect of attentional mechanisms. While we have previously used competitive activity distribution between columns as the basis of a theory of neocortical dynamics [Reggia et al, 1992], and also as a control mechanism for non-neurobiological cognitive/AI models of print-to-sound transformation and diagnostic problem solving, this will be the first time that it will be used as part of an attention mechanism based on thalamocortical interactions.

Finally, for a situated cognitive architecture to function effectively, it must be able to process events as they unfold sequentially in time. Processing of temporal/sequential events is supported within our framework by recurrent intra-region connections and recurrent inter-region pathways. This recurrent connectivity with its inherent delays leads to attractor states that are generally not fixed points, i.e., to quasi-periodic and chaotic attractors, and to switching between such attractor states as the basic mechanism for cognitive operations over time.

D. Learning: A Developmental Approach

The ability to learn is a critical aspect of human intelligence and thus a fundamental part of our theoretical framework. The needs in this area are extensive. Learning is required across a range of levels, from low-level sensorimotor processing and control through high-level cognitive functions and executive decision making, and across a range of contexts (supervised, reinforcement, and unsupervised scenarios) and modalities. We address these needs by integrating multiple learning algorithms in our framework, some of which are off-the-shelf methods and others of which are innovations that address specific needs. These algorithms act at different levels of our structural hierarchy, from individual cells and their connections to entire regions and their inter-regional pathways. The functional operations acquired by an initially generic region during learning are based on that region's unique position in an architecture's network as well as its intrinsic properties, just as is postulated to occur for functional localization in the cerebral cortex [Passingham et al, 2002]. As we explain below, the modular nature of our architecture allows learning to proceed in a multi-stage, incremental fashion that we refer to as a *learning agenda*. This approach is inspired by human neurobiological and cognitive developmental stages, and makes the training of a large scale cognitive system tractable. We now consider some of the details of the learning mechanisms, starting with the most conventional.

As with activation dynamics, at the level of cells and their connections we incorporate a variety of existing learning methods within our framework that are not intended as innovations of this work. These include unsupervised methods such as Hebbian learning, reinforcement methods such as temporal difference learning [Sutton & Barto, 1998], and supervised methods such as contemporary versions of error backpropagation like RPROP [Reidmiller & Braun, 1993] and methods for learning with recurrent networks. However, even at this lowest level we adopt some non-standard methods to address the broad range of learning methods needed by a general purpose machine intelligence, and give two examples of these here.

First, as noted earlier, processing temporal events is a fundamental requirement for a situated autonomous/semi-autonomous cognitive agent. At a minimum, the ability to learn to both recognize and generate temporal sequences is needed. There are a variety of effective supervised learning methods for temporal sequences, but unsupervised methods for distributed representations are much less developed. For the latter, recent discoveries of temporally asymmetric Hebbian learning in neocortex and other brain structures [Bi and Poo, 2001; Markram et al, 1997] have led to suggestions that this may be an important mechanism for learning temporal sequences [Rao and Sejnowski, 2000]. We recently created a specific implementation of temporally-asymmetric Hebbian learning and used it successfully with recurrent neural networks to "discover" an effective distributed representation for different temporal sequences of phonemes representing words [Schulz & Reggia, 2004]. This approach should generalize to analogous sequential tasks (e.g., learning to recognize an opponent's strategies). Our more recent experiences integrating and adopting sequence processing methods in larger, system-level models are encouraging, as we describe below.

A second non-standard approach to learning at the level of cells that is incorporated into our framework is the learning of activation dynamics. Most neural network learning methods assume an a priori, fixed activation dynamics that is at least loosely modeled after how individual neurons process information, with learning occurring primarily by changing weights on

connections. However, since the atomic units in our framework (cells/voxels) are not restricted to behave like individual neurons as long as they retain local information processing, our approach permits the activation function of cells (as well as connection weights) to be learned. For example, cells can learn novel ways to combine their individual inputs (rather than just as a linearly weighted sum), internal parameter values, whether to distribute their output activity in the usual non-competitive fashion or in a competitive fashion, and so forth. We have previously used this approach successfully in simple networks [Grundstrom & et al, 1996], and believe that it will generalize readily to the neural architecture described here, greatly increasing the flexibility and effectiveness of learning.

Learning at higher levels in the structural hierarchy, such as learning activity and weight values at the level of regions and their pathways, is largely unexplored in past neurocomputational systems. We believe that reinforcement learning methods are very promising at this level. In addition, fMRI data may provide useful guidance for setting pathway parameters such as the functional connectivity between regions. By *functional connectivity*, as opposed to structural connectivity, we mean the dynamic relationships between regions that exist during cognitive tasks. These relationships are associated with the covariance of regional activities as observed during functional imaging and often represented using structural equation modeling. Our initial attempts to guide task-specific pathway gain learning using fMRI data have been encouraging and are described below.

Finally, from a more global perspective, our framework recognizes that one cannot assemble a large-scale neurocognitive system all at once and simultaneously learn everything that is needed in one step. Thus, a central aspect of our methodology is that it incorporates a *developmental approach* that leverages our framework's inherently modular architecture. This is inspired by developmental processes shaping the human brain during childhood. Different brain systems have distinct developmental time courses, with synaptogenesis and synaptic elimination reaching peaks at different ages for different systems [Neville and Bavelier, 2000]. Behaviorally, children go through a sequence of stages in which psychological competencies appear in a fairly typical order, and these stages are loosely correlated with developmental changes in the brain [Kagan and Baird, 2004]. For example, children learn to recognize some aspects of phonemes of their native language well before they learn to produce spoken phonemes [Vouloumanos and Werker, 2004]. Our intent is not to model accurately the details of human childhood development, but to use this natural multi-stage process as a guide in assembling a large-scale neurocognitive architecture.

In our framework, the practical implementation of a developmental approach takes the form of a *learning agenda*. A learning agenda specifies a plan or procedure for the incremental construction and training of parts of a system, their assembly and further training, and so forth, until a complete and fully trained architecture is achieved. In a sense, this is a specific instantiation of a long-standing philosophy of how to go about creating a general machine intelligence that can be dated back to the early days of AI [Turing, 1950] and that continues to have its advocates today. This philosophy argues that one should initially aim to produce a machine intelligence with the abilities of a young child, and then allow such an artifact to learn additional abilities.

E. Need for a Hybrid Architecture

So far we have presented a conceptual framework for implementing a large-scale neurocognitive architecture based on modeling the hierarchical and modular organization, dynamics and plasticity of the human brain. Our emphasis has been on the neocortex and its interactions with subcortical structures that are most closely related to problem-solving, learning and cognition in general. We now turn to considering the requirements of a large-scale neurocognitive architecture. The core of our long-term approach remains focused on creating a network of neuromorphic regions as described earlier. However, for the short term of the next few years, it is likely that optimal results will be obtained by using a hybrid design that also includes symbolic methods from AI/cognitive science and control processes from the field of artificial life. We accordingly propose a three-tiered architecture that integrates these different methods, and describe a computational study of a prototype "mini-Roboscout" based on this architecture. We also examine the implications of some non-standard computational methods for developing a neurocognitive agent. This examination includes computational experiments assessing the effectiveness of genetic programming as a design tool for recurrent neural networks for sequence processing, and experiments measuring the speed-up obtained for adaptive neural networks when executed on a graphical processing unit (GPU) rather than a conventional CPU.

Our specific architecture, involving repetitive use of generic neural components and multistage learning, should facilitate highly parallel processing, robustness to damage, and eventual physical realization in fine-grained parallel processing architectures. We believe that this approach, or something very much like it, will ultimately be successful in creating a general-purpose machine intelligence. However, uncertainties in contemporary knowledge about brain functions, and in our understanding of how to capture some aspects of cognition in neural algorithms, raise the question of what the optimal strategy is for achieving such an architecture. This is a critical question if one plans to gauge success by the ability of a developing cognitive architecture to function in naturalistic settings within the short period of a few years.

Our answer to this question is that trying to implement a full-scale, purely neuromorphic architecture immediately and all at once would be extremely difficult and carry a high risk of failure. A much better approach over the short term would be to develop a *hybrid architecture* that combines neurobiologically-inspired methods and cognitively-inspired methods within a single unified framework. By "cognitively inspired methods", we mean more conventional symbolic and numeric methods from cognitive science and AI rather than neural computation methods. In such a hybrid architecture there is a third "dimension of integration" in addition to the behavioral tasks and cognitive mechanisms that we described earlier. This *computational methodology dimension* of integration refers to combining the variety of computational methods that are available today for producing various aspects of machine intelligence. At one end of the spectrum are the cognitively-inspired methods that have dominated cognitive science, AI, and related fields. They are often referred to as "top-down" approaches and include symbolic methods such as first order predicate calculus, production systems, and heuristic search as well-as statistical pattern classification techniques. At the other end of this spectrum are neural computation approaches inspired by the brain that form the basis of our theoretical framework, and biologically-inspired methods developed in the field of artificial life. For example, swarm intelligence methods for movement control are particularly relevant [Rodriguez, 2005]. These "bottom-up" approaches typically start with a distributed representation of information and

emphasize learning and self-organization, viewing cognition as a phenomenon that emerges from the dynamics of a neurobiologically-inspired complex system, not something that one explicitly programs in. The view presented here is that integrating these computational methodologies rather than restricting one's approach to just one class of methodologies is most likely to be productive over the short-term of the next few years.

There are at least two reasons for starting with a hybrid approach. First, past successful applications of neurobiologically-inspired and cognitively-inspired machine intelligence on focused, limited-scope tasks have largely been complementary. Neurocomputational methods have excelled at *learning* to do "low-level" tasks like pattern classification, autonomous movement control, and associative memory, plus they have demonstrated a robustness to damage/noise and an ability to generalize. In contrast, more traditional symbolic methods of cognitive science and AI have excelled at "high-level" aspects of cognition such as problem-solving, inference, planning and executive control. The point is that if these complementary neurocomputational and symbolic methods can be effectively integrated, the resultant combination would potentially be much more powerful than either methodology alone. The second primary reason for adapting a hybrid approach is that it automatically leads one to a roadmap for achieving the long-term goal of a fully neuromorphic machine intelligence, by proceeding as follows. Initially, implement a hybrid system with both neuromorphic and symbolic methods. Then, as knowledge of brain function improves and neurocomputational technology advances, gradually replace functions captured in the more traditional symbolic components of this architecture with neuromorphic components. At any time in this process, the remaining cognitively-inspired components effectively define the critical research agenda for achieving a full-scale purely neuromorphic architecture: this consists of the aspects of cognition that remain implemented in the symbolic framework because their neuromorphic implementation remains undefined. In the following we extend our theoretical framework to encompass cognitively-oriented symbolic and other methods in a unified setting and present a roadmap for the development of a full-scale architecture within this extended framework.

F. Design and Implementation

Figure 3 provides an overview of our proposed three-tier architecture. The *sensorimotor level* interacts most directly with the environment, the *cognitive level* is the heart of the system, and the *executive level* captures "executive functions" as that term is used in neuropsychology. This three-tier structure is directly inspired by the organization of the human nervous system. The sensorimotor level corresponds not only to subcortical structures, but also to primary sensorimotor neocortex and to neocortical regions dealing with automatic sensorimotor transformations (e.g., Brodmann area 7a) and inter-modal sensory transformations. The cognitive level corresponds to neocortical regions often referred to as "association cortex", such as language areas, substantial portions of parieto-temporal cortex, etc. The executive level corresponds to prefrontal cortex, anterior cingulate gyrus, and related regions. Of course, this partitioning of brain regions and functionality is imprecise and ignores some overlapping of functionality (for a review of this issue, see our earlier report [Tinerella et al, 2006]).

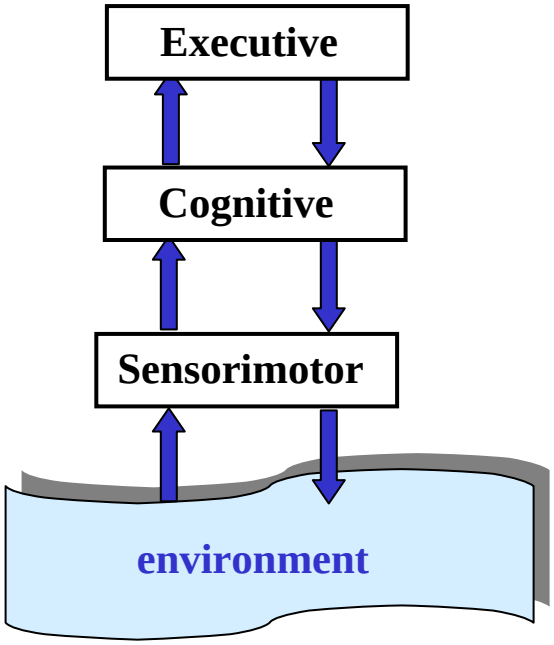


Figure 3: The three-tier organization of our proposed neurocognitive architecture.

Independent of its neurobiological inspiration and correlates, the three tier organization of Figure 3 provides a powerful organizing framework for a cognitive architecture, as follows. The sensorimotor level, which could be implemented and function in isolation of the other levels in an environment, captures functionalities whose execution is largely automatic or "reflexive". These include basic pattern recognition, sensorimotor coordinate transformations, and elementary actions such as moving through an environment having obstacles or executing an arm control command. This level is also the appropriate place for responses to environmental events that must be responded to very quickly, such as the immediate recognition of dangerous situations and reflexive behavioral actions.

In contrast, the cognitive level encompasses so-called "higher cortical functions" such as language, deduction, cause-effect reasoning, problem solving, motor program selection, etc. Mechanisms at this level, when coupled with the sensorimotor level and even in the absence of the executive level, should make an agent taskable, i.e., able to carry out specific albeit relatively simple goals in response to a command. Episodic memory is also available at this level, permitting an agent, for example, to recall a recently followed route and the objects it observed along the way. While slower than the sensorimotor layer, much of the processing at this cognitive level is still fairly automatic/deliberative, e.g., the recognition of a sequence of phonemes as corresponding to a specific word. With just the cognitive and sensorimotor levels present, although an agent could follow commands, it would have no ability to judge the appropriateness of its goals, to make elaborate hierarchical plans for achieving those goals, or to understand its own reasoning status.

The third, highest executive level of the architecture carries out "executive functions" by interacting with the cognitive level. It is able to generate new goals and subgoals to be followed

by the cognitive level (and thus also the sensorimotor level) based on information about its current situation. This is the most reflective and potentially slowest part of the overall architecture. It is responsible for generating plans (goal sequences), for monitoring execution of those plans, and for generating revised plans (re-planning) when unanticipated events occur. It is also at this executive level that social intelligence appears (theory of mind) and at which inferences can be made about the agent's own state and reasoning processes (metacognition).

The three-tier architecture we are presenting also provides a fairly natural organizing framework for combining different computational methodologies in a single system. Current neurocomputational and artificial life methods are fairly effective at the sensorimotor level, especially relative to symbolic methods in AI and cognitive psychology. In contrast, symbolic methods currently are more effective at the executive level. In between, at the cognitive level, all of these methods have a role to play. What becomes critical is the need for modularity in the components that form the three levels. Modularity is important for information hiding, including the nature of computations inside of a module, to allow a clear integration of different computational methods in a single, full architecture. Modularity is also critical to a rational roadmap to implementation, as we explain in the Roadmap section later in this report.

Our discussion so far has focused only on how the basic three-tiered architecture of Figure 3 functions in its environment, managing problem-solving and carrying out tasks. We refer to this aspect of the agent as its *basic operation*. An important question in this context is how functions such as memory, learning and attention are to be accommodated within this framework. The answer is that these kinds of functionality span and are essentially orthogonal to the three "horizontal" levels of sensorimotor, cognitive and executive functions. Their "vertical" nature indicates that, like the agent's basic operations, their functionality involves all three levels of the core architecture, providing an overall matrix organization.

For example, in terms of memory, at the sensorimotor level weight matrices store inter-modal sensory transformations and simple motor programs, the latter in recurrent networks. The cognitive level encompasses semantic associative knowledge, a lexicon, and episodic plus working memory. The executive level maintains a memory store that includes a current goal/subgoal stack and recollection of the status of specific previously encountered external agents (friend, foe, mentor, competitor, etc.). Likewise, *learning* that alters the contents of memory is distributed across the three functional levels. Finally, attention and control mechanisms operate concurrently at all three levels, an organization that is a natural fit to the three main classes of attentional mechanisms recognized by many cognitive psychologists (reviewed in [Raz & Buhle, 2006]). More specifically, bottom-up *alerting* to external stimuli associated with subcortical brain regions roughly corresponds to our sensorimotor level, bottom-up (exogenous) and top-down (endogenous) *selection/orientation* to specific sensory modality events associated with parietal lobe activation to our cognitive level, and *executive attention* (dealing with conflict identification and metacognition) associated with anterior cingulate cortex activation to our executive level [Raz & Buhle, 2006].

G. Implications of Non-Standard Methods

The highly parallel nature of neural computations, and the potential parallel implementation of symbolic and other cognitive/AI algorithms, raises the issue of how the processing in our neurocognitive architecture can take advantage of parallel processing in non-standard computer

architectures. In the following, we first consider high performance computing systems that are currently available, and then we examine some longer term possibilities such as nanotechnology, quantum computing, and using evolutionary computation as a design aid.

1. High Performance Computing

The brain employs massive parallelism to allow it to perform complex calculations in real time. Artificial systems of sufficient complexity also face significant challenges in producing real time performance, and these systems could potentially benefit from using massive parallelism too. For this reason, we consider here the possible use of parallel computation in our hybrid reasoning system. We focus on performing neural net computations on parallel hardware. These neural computations map more naturally into parallel systems than the more inherently sequential processes in the non-neural components of our architecture.

Different types of neural networks present different challenges to parallel processing. Most importantly, the type of interconnectivity in a network can influence the amount of communication required between different processors. This can vary significantly between feed-forward and recursive neural nets. Communication between neural units is typically one of the chief bottlenecks in parallel processing. None-the-less, significant speedups have been reported by using parallel hardware to implement multi-layer perceptrons ([Long and Gupta 05; Pethick et al.; Seiffert 02]. For example, [Long and Gupta 05] report experiments on both a 160 node Beowolf cluster, and on a system containing 500 Intel Itanium processors. They report that they are able to maintain constant run times as the number of neural units and the number of processors both scale linearly.

While a variety of hardware platforms have been considered for parallel implementations of neural networks [Zhu and Sutton 03], [Seiffert 02] argues that typically clusters, such as a Beowolf cluster, are the most practical choice due to their wide availability and good performance/price ratio. In fact, one of our goals is to explore parallel implementations of neural networks for clusters that use efficient, off-the-shelf hardware. For example, Graphics Processing Units (GPUs) offer exciting potential for high performance computing because their use in 3D consumer games is driving a dramatic increase in their performance/price ratio. Some estimates describe a 2.8-fold annual growth rate in processing power, compared to a Moore's law rate of 1.7 per year. Because of this opportunity, general purpose programming tools are springing up for GPUs, including linear algebra libraries and high level programming languages.

In part due to interest in using neural nets to control game characters, there have already been implementations of neural nets for GPUs. [Bernhard and Keriven 05] describe an implementation of a neural net with spiking neurons on a GPU. This includes a general purpose spiking neuron simulator. They obtain speedups of up to a factor of twenty compared to a comparable system running on a CPU. [Oh and Jung 04] obtain comparable speedups in their implementation of multilayer perceptrons on a GPU. At the same time, implementation on a GPU is more difficult, because operations must be mapped onto vertex and pixel shading operations, which the GPU can then perform in parallel using its 24 pipelines. However, the growing availability of general-purpose libraries such as [BrookGPU] should ease this process. To our knowledge, neural network systems have not yet been implemented on a cluster of GPUs. This offers the potential for tremendous speedups, but also raises a number of challenges. These include finding effective ways to partition the computation in a scalable way across a large

number of GPUs. The modularity of our brain inspired architecture should make this process somewhat easier, however. Some preliminary steps are summarized later in a pilot study we undertook as part of this work to evaluate neural network simulations run on GPU processors.

2. Nanotechnology and Quantum Computing

Nanotechnology refers to the fabrication processes and device structures used to build transistors or circuit elements that are smaller than roughly 100 nanometers in size. For example, the insulating gate oxide in the state-of-the-art metal oxide semiconductor field-effect transistors (MOSFETs) consists of only five atomic layers, which add up to only 1.2 nanometers in thickness [Tyagi et al, 2005; Jan et al, 2005]. The gate length has reached 35nm in high-volume microprocessor manufacturing on 300mm wafers. Downsizing has been successful in increasing the density and enhancing performance of integrated circuits. With new fabrication methods, strained silicon, low- and high-k dielectric, surround-gate structure, etc., industry continues to develop new downscaling recipes and to pack more transistors on a chip. However, even before the laws of physics set a clear, hard limit, heat dissipation imposes a practical size limit. There are two main applications of nanoscale MOSFETs: CPU and memory, and the most advanced chips contain more than one billion transistors. The high-speed operation of the CPU inevitably results in high power consumption. Today, conventional power dissipation technology sets a limit at about 100Watts [Ravi et al 2005]. Because of this heat dissipation problem, Intel officially abandoned their effort to boost up clock frequency to 4GHz, and is beginning to look into multiple-processor and parallel computing as an engineering solution.

While the IC industry continues on the path of downscaling for commercial products, researchers are also looking for solutions. The ultimate downsizing of transistors is expected to be to only nanometers, and operating principles will face a fundamental change. As opposed to classical diffusive transport, quantum phenomena, including single electron charge, size quantization, electron wave-like interference, and ballistic transport, are expected to dominate transistor characteristics. New classes of materials, such as semiconductor nanowires, carbon nanotubes, and even molecules are being considered and investigated. The current thinking is that these alternatives could work in conjunction with conventional circuits and that a hybrid chip can, for example, deliver both high speed computing and high volume memory. Note that these potentially lower-power alternatives must still obey the laws of physics, including the same thermal dissipation issue. We will potentially be able to utilize the remarkable new properties of these quantum-based transistors in computing. The main anticipated difficulty is that quantum-based transistors operate without dissipation, for otherwise phonon emission destroys the quantum state of electrons. Although appealing for high-speed, low-power switching, it remains to find ways to cascade many stages in series for practical applications.

The use of nanotechnology in neuroscience is an emerging research area, with current work examining technologies that can interact with neurons and glial cells at the molecular level, advanced imaging and manipulation of neurons using functionalized quantum dots, and approaches to supporting functional neural regeneration following nervous system trauma (reviewed in [Silva 2006]). Much less has been done with nanoelectronics for artificial neurocomputational systems. In spite of the continual advancement of nanotechnology, forming interconnects even with 3D integration is still the primary obstacle in circuit implementation, and the interconnect requirement will ultimately limit the number of neurons and the functionality of conceivable neuron-chips. There is one apparent advantage to developing such new

nanotechnologies around silicon: there is already an existing infrastructure.

On the other hand, quantum computing is a totally different approach [Nielsen & Chuang 2000]. Unlike a classical bit, the qubit (quantum bit) is a superposition of the two states |0> and |1>. Quantum mechanical operations, such as superposition of two eigenstates, entanglement of two qubits, unitary transformation, logic gates, etc., are designed to perform operations on these qubits. Though quantum computing is still in its infancy, extensive experiments have been carried out and demonstrated the validity of the quantum computer concept. However, a full-blown quantum computer is many years away. The recent development and funding of quantum hardware implementations is actually driven by proposed applications in factorization, teleportation, encryption, and sorting algorithms. New algorithms in computing and information processing are still being investigated. At this moment, there is strong competition in developing a semiconductor-based qubit, preferably in silicon. A single electron confined in a semiconductor quantum dot, under the influence of a magnetic field, forms a two-level system that is considered ideal as a qubit. The practical difficulty is to engineer such a quantum dot and perform experiments to manipulate and measure the electron spin. Multiple operations, estimated to be of the order of 10^6, must be done before the electron spin looses its spin phase coherence. Any breakthrough in constructing a qubit in any material system that can satisfy several basic requirements, i.e., long coherence times and up-scalability, will have a large impact on the computing community. Inspired by parallelism, which is an aspect of quantum computing by default, applications in artificial neural networks have been proposed in the areas of classification, associative memory, image processing, and pattern recognition. As in the case of developing a classical computer, feasibility has to be determined by the characteristics of the physical properties of the qubit and quantum logic gates. One fundamental problem to be addressed first is that quantum mechanics is linear, but neural networks generally involve nonlinear effects. The discrepancy might be solved by allowing a qubit to interact with its environment and to be subject to a time-varying external perturbation. This is a research field in its infancy. The parallelism nature of quantum computing should be fully exploited for a clear understanding to the potential impact in artificial neural networks.

3. Genetic Programming

Evolutionary computation refers to a set of general-purpose search algorithms inspired by natural selection and evolution [DeJong 2006]. These algorithms use a population of individuals that represent potential solutions for a given problem. During each generation, the environment (via a fitness function) indicates which individuals/solutions are more fit than others, and the next generation of the population is produced by selecting the most fit individuals and modifying them via mutation, recombination, and/or other genetic operators. In this way, starting from an initial randomly-generated population that usually represents poor solutions to a problem, progressively better solutions are identified over time.

There are a variety of different approaches to evolutionary computation, including genetic algorithms, evolutionary programming and evolutionary strategies. Each approach, in its canonical form, has its own representation scheme and genetic operators, as well as different philosophies/details to the simulated evolutionary process. For example, considering just the representation of the genetic encoding, genetic algorithms use binary strings, evolutionary programming uses finite state machines, and evolution strategies use real-valued vectors. The approach we focus on here is called *genetic programming.* Genetic programming (GP) literally

refers to the evolution of computer programs, but is also often taken to mean evolution of data structures represented as trees [Banzhof et al, 1998]. Mutations typically are formulated as replacement of a randomly-selected subtree with a new, randomly-generated subtree; crossover is implemented as the swapping of subtrees between two individuals. As with genetic algorithms, most workers in GP take crossover to be the primary genetic operator.

During the last several years, there has been increasing use of GP, as well as other evolutionary computation methods, as a creativity tool in design problems. For example, GP has been used to evolve patentable electronic circuits, antenna configurations, novel pilot combat maneuvers, music, and robotic mechanisms. For example, our own group has used both genetic algorithms [Lohn & Reggia 1997] as well as GP [Pan & Reggia 2006] to evolve rules that produce self-replicating structures in cellular automata environments. Two key points come out of all of this design-oriented GP work. First, GP tends to discover solutions to problems that are creative in the sense that they are quite different than what human designers produce. Sometimes the solutions are substantially better than past human solutions. Second, the use of GP can be viewed as a type of machine learning. In essence, GP involves a fitness-guided search through the space of potential designs for a problem, learning which designs are most effective as it goes.

Given the recent progress in using GP as a design aid, a natural question is whether GP might be adopted to create/discover novel aspects of a neurocognitive agent. We focus on the neural components in the following, but the applicability may actually be broader than just that. Neuroevolutionary methods have been used to create a substantial range of interesting neural network designs (e.g., see [Yao 1999] for examples). Two aspects of this past work are most relevant to biologically-inspired cognitive architectures. First, *developmental representations* of genetic material have been devised that specify how to "grow" a neural network (the phenotype) rather than directly encoding its structure. Examples include graph generation grammars and cellular encodings [Gruau 1996]. In addition to incorporating a model of the biological process of neurodevelopment, developmental representations let one represent individuals as a tree (the genome) while evolving general-graph structures as neural networks. Second, growing attention has been given to producing neural systems having *modular architectures*. Work in this area has been inspired in part by recognition that biological nervous systems are highly modular and hierarchical. For example, the vertebrate cerebral cortex is composed of cortical columns (small modules) that are in turn components of functional regions (large modules) that collectively form the cerebral cortex.

Combining modular design with developmental encodings, and integrating GP evolution of neural network architectures with more conventional neural network learning via synaptic weight changes, seem especially promising avenues to explore. In the results below, we describe a pilot study evolving recurrent neural network modules to examine this hypothesis.

Results

We now describe the results of exploratory computational experiments that we completed to establish the plausibility of the concepts introduced above and to examine their implications.

A. Associative Word Learning Model

Our first experiment focused on the assembly of an *associative word learning model***.** This model is based on the "classic" and highly influential Wernicke-Lichtheim-Geschwind (WLG) theory of language processing that is widely known in neuropsychology and clinical neurology [Brown & Hagoort, 1999]. Accordingly, it differs from most past computational models related to language in that past models have largely simulated the cognitive processes involved and generally did not intend to represent the underlying cortical regions and their interregional connections explicitly. We use the model to address two specific questions. First, beginning with an untrained model consisting of interconnected neocortical regions spanning both cerebral hemispheres, is it possible to create a left-lateralized computer simulation of the primary regions and pathways of the WLG theory that can learn to recognize heard words that are object names, repeat them, and associate them with the appropriate objects? Second, assuming that one can successfully implement a computational simulation of the WLG theory, to what extent does it behave in ways reminiscent of the classic aphasia syndromes following focal cortical lesions?

The basic functional-anatomic framework of the WLG theory is illustrated in Figure 4. Broca's area (BA) and Wernicke's area (WA) are the most prominent language processing centers in almost all theoretical models of language processing, including the WLG theory. Although the relative importance of these areas to different language functions is a long-standing question, there is no doubt they each serve separate roles. WA receives input from primary auditory cortex (A1), among other areas. The language deficit known as Wernicke's aphasia is closely associated with WA loss, and is characterized by impaired comprehension and repetition ability, but with some spared ability to produce fluent, but often meaningless, verbal utterances. In contrast, BA is believed to be responsible for more expressive aspects of language, playing an important role in grammatical speech production. Destruction of BA along with surrounding frontal cortex is associated with Broca's aphasia, an impaired ability to produce linguistic output despite retained comprehension. The arcuate fasciculus (AF) is the pathway connecting WA and BA. There are some linguistically impaired patients, said to have conduction aphasia, who are capable of both comprehending and producing speech, but incapable of repeating heard words. Historically the proposed underlying deficit in these individuals is blockage of the AF 's "conduction" of information from WA to BA. Currently it is believed that communication between these areas is mediated by more extensive anatomical routes, including regions such as the supramarginal gyrus in the parietal lobe. Another parietal area, the angular gyrus, appears to play an important role as functional center of linguistic and visual object comprehension as part of a distributed semantic system. Lesions to the angular gyrus and surrounding cortex have been shown to lead primarily to multimodal comprehension deficits and have been classically associated with transcortical sensory aphasia. Finally, inferior cortical areas have been linked with recognizing and naming visual objects (confrontation naming). Object recognition is

believed to take place through a ventral visual pathway, leading from V1/V2 to inferior temporal cortex (IT), with IT representations being more complex and not retinotopic. While classical WLG theory did not include IT, lesions along this pathway lead to loss of object recognition.

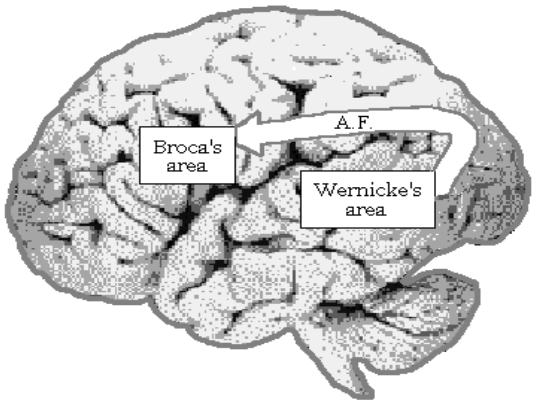


Figure 4. Central aspects of the Wernicke-Lichtheim-Geschwind theory. Wernicke's area (receptive processing) is connected via the arcuate fasciculus (AF) to Broca's area (expressive processing). Inferior parietal regions such as the supramarginal gyrus and the angular gyrus are viewed as important tertiary association cortex but are not labeled here. The cortical areas supporting language are assumed to be only present in the dominant left hemisphere.

While the WLG theory is clearly inadequate to account for all language phenomena and it does not incorporate some important concepts from contemporary psycholinguistics [Caplan, 2003; Poeppel and Hickok, 2004], it is a powerful organizing heuristic for understanding the neurobiological basis of language that has influenced most contemporary theories of language. To our knowledge, no one has previously developed a neurocomputational model of language functions based on this traditional neurological theory.

The architecture that we assembled is illustrated in Figure 5 and consists of a network of perisylvian cortical regions forming the core of the WLG theory. This implementation is also informed by the results of studies over the last few decades that were not available to the founders of the WLG model, such as functional imaging. Some of these regions, their activation dynamics, and their learning procedures are inspired by earlier, simpler neural network models. Broca's area (BA) and primary motor cortex (M1) are modeled after an earlier phoneme sequence generation model [Reggia et al, 1998], while primary auditory cortex (A1) and superior temporal gyrus (our rendition of Wernicke's area (WA)) are modeled after an earlier phoneme representation model [Schulz & Reggia, 2004]. The remaining four regions (visual cortex (V1/V2), inferior temporal cortex (IT), and supramarginal gyrus (SMG), and angular gyrus (AG)) use similar methods. Each region is a cellular array in the sense defined earlier, and there are recurrent intra-regional connections that are not shown in Figure 5.

While the classic WLG model is generally used to describe human left hemisphere language processing pathways only, more recent research has suggested that homologous right hemisphere processing circuits may also exist and contribute to right hemisphere language processing. Experimental observations suggest that both hemispheres have substantial *potential* for language processing initially, with (usually left) hemispheric specialization for language arising during childhood development and language acquisition. In this context, our computational model's structure includes two initially identical hemispheres. For each left hemisphere region, there is a homologous right hemisphere region homotopically connected to it via simulated corpus

callosum connections. The exception is that only a single M1 output layer is present, with connections back to both left and right hemisphere BA areas. This ensures that only a single output is produced based on the input received from pathways of both hemispheres. Thus, there are a total of 15 simulated cortical regions in the model. Except for different random initial weights, homologous left and right regions are initially identical. In effect, two identical sets of mirror image hemispheric regions are present, with one being designated the left hemisphere and the other designated the right. The challenge is for left hemisphere dominance, an important explicit feature of the WLG framework, to emerge during learning even though both hemispheres receive the same input patterns. Our intent here is not to suggest that paired left and right hemispheric regions are ultimately necessary in a neurocognitive architecture; we are only trying to determine whether current methods for guiding which functions become acquired by which regions can scale up to a model of this size and complexity.

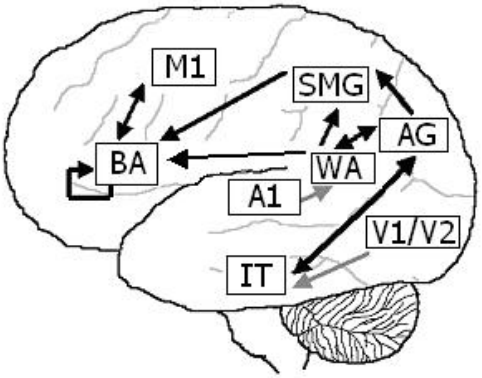


Figure 5. Modules within the associative word learning model's left hemisphere, with arrows representing inter-region pathways. Grey arrows indicate unsupervised learning pathways, and dark arrows indicated supervised learning pathways. Intra-regional recurrent connections exist but are not shown. Homologous regions and pathways are present in the right hemisphere but not pictured here. BA = Broca's area, WA = Wernicke's area, IT = inferior temporal cortex, SMG = supramarginal gyrus, AG = angular gyrus.

We limited the model to processing single word names for tractability, and because it is an important part of routine "bedside testing" in clinical neurology. Inputs to the model are fifty "heard words" taken from the NetTalk corpus represented as temporal sequences of auditory phonemes in primary auditory cortex (A1), and images of objects from the Snodgrass-Vanderwart corpus (Snodgrass & Vanderwart, 1980) in primary visual cortex (V1). Input of a spoken word was done by presenting its phonemes as a temporal sequence of patterns imposed on A1. Input patterns are presented simultaneously to A1 areas in the left and right hemisphere. Each individual input phoneme is encoded as a unique distributed pattern of 34 auditory distinctive features (voicing, duration, nasality, etc.), normalized to unit length to prevent input patterns with many features from dominating learning. Visual input consists of 50 two-dimensional images, each corresponding to one of the words described above. These images were taken from the line drawings of familiar objects in the Snodgrass and Vanderwart (1980) corpus, converted and scaled to a 50 x 50 bitmap format. Outputs from the model are "spoken words" represented as temporal sequences of motor phonemes in primary motor cortex (M1) corresponding to the correct pronunciation of the given input word or picture. Each motor phoneme is encoded as a pattern of 20 articulatory distinctive features (using a different encoding than A1), so each neural element in M1 represents an articulatory distinctive feature.

During training, the model produces a sequence of output phonemes, ideally the same number as the number of phonemes in the target output, plus an output vector of all zeros called the stop phoneme and designated /#/. No specific functionality is assigned a priori to model cortical regions, other than that implicitly present due to their location and interconnectedness in

the network. Initially, homologous cortical regions in the simulated left and right hemispheres are symmetric except for randomly assigned synaptic weights, so before training both hemispheres contribute equally to output and the model structure does not favor either left or right hemisphere specialization.

Rather than trying to train all of the model's functions simultaneously, we adopted a learning agenda that consists of three stages. The goal of the first stage is to develop representations within the primary sensory association areas (IT and WA) using unsupervised learning. This phase corresponds to attentive viewing and listening to pictures and auditory stimuli without producing output, much as an infant experiences both visual and auditory stimulation following birth before language production occurs [Vouloumanos, Werker 2004]. Using the 50 stimuli described above as inputs, with each stimulus having both a visual (image in V1/V2) and auditory (temporal sequence of phonemes in A1) representation, learning proceeded separately for each stimulus modality. Each iteration consists of the stimulus information being set in the sensory cortical area (V1/V2 or A1), activity propagating to the sensory association area (IT or WA), and finally weights being adjusted using competitive Hebbian learning based on activity within the sensory association areas. For heard words presented as a temporal sequence (e.g., for kite, /k/, /ai/, and /t/ plus stop phoneme), for any given auditory stimulus multiple inputs are received by the system, with learning occurring after each input phoneme using temporally-asymmetric Hebbian learning [Schulz, Reggia, 2004].

The goal of the second stage of training is to learn the bidirectional associations between word representations in WA and image representations in IT. This was accomplished using resilient error backpropagation [Reidmiller and Braun, 1993] where area AG served as a "hidden layer" between WA and IT. While error backpropagation is generally viewed as a form of supervised learning, note that the model is free to determine any representation (i.e., encoding) for the word-image associations that it learns in area AG.

The goal of the third stage of training is to have the model generate the correct output sequence of motor phonemes to name a seen picture or to repeat a heard word. Learning to repeat a heard word is especially challenging: generation of the output motor phonemes does not start until *all* input auditory phonemes for that word have been processed. Thus, the model must discover an internal representation for each temporal auditory sequence that persists and is adequate to generate the correct corresponding temporal sequence of motor phoneme features. Learning during this second phase occurs for all connections to and from areas SMG, BA, and M1 using resilient error backpropagation [Riedmiller and Braun 1993].

Hemispheric specialization is an important aspect of the WLG model. Past computational studies using simpler models than the one we are studying here have found that lateralized functionality can be consistently produced during learning when corresponding left and right cortical regions are asymmetric in size, excitability or synaptic plasticity [Shkuro, Glezer et al. 2000; Reggia, Goodall et al. 2001; Weems and Reggia 2004]. We elected to encourage left hemisphere specialization in our model by giving the left hemisphere a learning rate advantage throughout training (all three phases). Thus, while the two hemispheres were structurally identical and connected through a simulated corpus callosum for each area, the left hemisphere was a more rapid learner and therefore expected to become a better language processor.

We adopted four performance measures to assess model behavior. *Repetition* measures the percentage of correct output phonemes produced following presentation of auditory input

words. *Naming* is measured in the same way as repetition, except that it reflects percent correct phonemes following visual stimuli. *Fluency* is a measure of the percentage of the expected number of phonemes that are produced following auditory input (unlike with repetition, the correctness of the phonemes produced is not considered). Finally, *recognition* is a measure of the number of correctly identified stimuli, regardless of the correctness of phonemic production. The angular gyrus has been identified at times as the location for storage of semantic information [Caplan 2003; Dronkers, Wilkins et al. 2004] and as a modality-independent association area [Binder, Frost et al. 1997; Booth, Burman et al. 2002]. In our model, it is the earliest area to receive information from both visual and auditory modalities, and thus is in a unique position to associate information received through these two stimulus input pathways. For these reasons, we defined recognition to be the extent to which the AG regions' activation patterns bilaterally, following a stimulus, could be used to determine correctly what the stimulus name had been. A value of 100% correct on this measure with the intact model implies that a *unique* activation pattern was created during learning in the AG's for each word in the training data. Following lesions to the WLG model, the value of this measure indicates the extent to which the original representations of learned words in the intact WLG model persist in the lesioned WLG model.

Model performance, as determined by our four performance measures, was assessed in ten independent simulations that were identical except for initially random weights. Figure 6 shows performance of the intact model before (a) and following (b) initial training. We see that the trained model performs nearly perfectly for each of the four dependent measures. Thus, the model developed unique internal representations (AG activity patterns) for the individual named objects. It was also successful in identifying the simulated visual and auditory input stimuli and mapping them onto the correct series of output phonemes. This is a substantial accomplishment, as the correct sequence of phonemes, ranging from three to ten in length, needed to be produced from two different forms of input based solely on learning synaptic connection strengths in a complex recurrent network. We also measured a laterality coefficient value [Shkuro et al, 2000] of -0.36, indicating that the left hemisphere was much more influential role than the right.

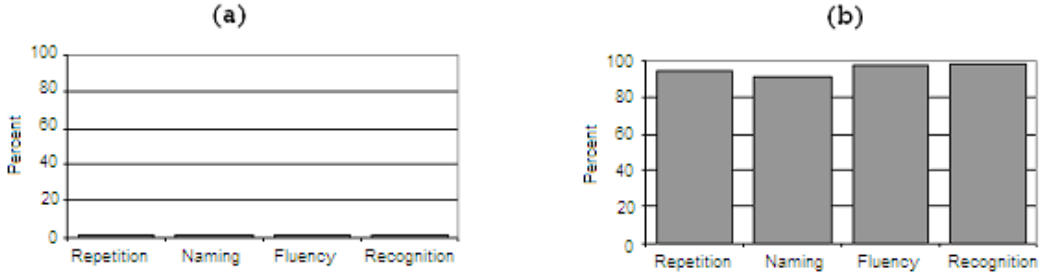


Figure 6. Model performance assessed using four measures. Before training (a), the model fails to identify or recognize the stimuli, but after training (b) the model performs consistently well, above 90% for each measure.

In addition to testing the intact model, we also examined model performance following simulated lesions to the regions WA, BA, AG, and IT, and to the AF pathway. Lesions consisted of "removing" 75% of the neural elements in a given area (or 75% of the connections of the arcuate fasciculus) by permanently fixing their output to zero. The lesions roughly correspond to damage *classically* associated with Wernicke's, Broca's, and transcortical sensory aphasia, visual anomia, and conduction aphasia, respectively, although correspondences between these biological lesion sites and aphasic syndromes are imperfect. Remarkably, simulated lesions to

the individual regions of the model generally produced deficit patterns reminiscent of the corresponding classical aphasia syndromes seen in people [Caplan, 2003]. For example, when the left hemisphere AF was damaged, both repetition and naming performance measures dropped below 40%. Fluency, although affected, dropped much less, and recognition ability did not drop at all. In contrast, damaging the right hemisphere AF had minimal effect on all four performance measures. Damage to the left arcuate fasciculus (AF) in humans is classically associated with impaired naming and repetition ability, but a retained ability to comprehend and produce some linguistic output [Anderson et al. 1999], consistent with the model's behavior. Comparable results were obtained with lesions to other model areas [Weems and Reggia, 2006].

To summarize, the key finding of3 the current model was that it is capable of learning "from scratch" the visual image, auditory phoneme sequence representations (names), and motor phoneme sequence representations of fifty separate objects. We consider such results to be promising. Remember that we did not assign any functionality a priori to any cortical region in the model, nor did we devise any new neurocomputational methods in creating the model (i.e., we used off-the-shelf modules, activation dynamics, learning methods, etc.). The learned ability of the model to produce output corresponding to the correct phonemic representation of both auditory and visual input stimuli is not trivial, as both the auditory and motor phoneme distinctive feature representations were distinct and complex; associations had to be made at several processing levels via multi-layered neural networks. For example, in word repetition, the model did not begin to generate output motor phonemes until after *all* auditory phonemes had been processed for that word, so it had to retain an internal representation of the word from which to generate its correct pronunciation. The model had to learn to not only map to the correct sequence of motor output vectors representing phonemes from the input patterns, in whatever form that input took (temporal auditory phoneme sequence or static image), but also had to know the correct temporal length of the appropriate output and cease output phoneme production at the correct time. This is considerably more complex than simple association learning, yet the model demonstrated near perfect performance on all performance measures in spite of the simplicity and small size of the cortical regions simulated relative to their biological counterparts. The model demonstrated patterns of word processing deficits following left hemisphere lesions much like those observed in human aphasic patients.

B. Delayed Match-to-Sample Model

Executive functions are the high level cognitive abilities that allow manipulation of information. One major component of executive function is the ability to keep information in a short-term memory so that it can be manipulated and combined with other information. Both single-cell recordings in animals and imaging studies in humans suggest that this ability, called *working memory (WM)*, involves a network of interacting brain regions, with the frontal cortex playing a key role. Decision-making is also thought to involve operations that require comparisons of alternatives that are held in WM, and is closely linked to WM. However, the neural underpinnings of these functions are poorly understood.

Although functional magnetic resonance imaging (fMRI) has revealed brain regions that are involved in WM, there still are no techniques for relating fMRI activity to underlying neuronal circuit properties. In order to understand how the operations of WM are implemented in the brain, we have developed a large-scale systems-level model of WM that includes a method for relating the neural mechanisms to human fMRI data. The goals for the model are that it be able

to perform WM tasks that are typically used in fMRI studies, that its neural dynamics mimic those found in animal single-cell recordings, and that it reproduce human imaging results quantitatively in the brain regions included in the model. This approach makes it possible to begin to explain the human data in terms of underlying neuronal circuit dynamics. The model is composed of multiple brain regions and includes a working memory circuit that maintains representations of recently seen objects in short-term memory, and it performs a delayed match-to-sample task, in which it makes a decision about whether there is a match between a stimulus held in working memory and a stimulus that is presented after a delay, possibly with intervening stimuli. An attentional system that models the presumed effects of dopamine controls the performance. The model fulfills three major requirements for a working memory system: 1) maintaining representations in short-term memory; 2) resistance to interference; and 3) the ability to make a decision that initiates an update of the contents of the memory. Together, these features implement a form of executive control that is necessary for intelligent behavior.

The basic model that we created addresses the delayed match-to-sample task, and incorporates the ventral visual pathways. In the delayed match-to-sample task, subjects are asked to determine whether or not the current input stimulus matches a previously seen stimulus that is retained in working memory. Previously we studied a simpler model in which inputs are visual patterns (letters, simple geometric shapes, etc.) similar to those used in human fMRI experiments. Outputs from the prefrontal region are decisions (match or no match) about whether the current visual input matches the previous one stored in working memory. Prefrontal cortex is believed to play a critical role in working memory during this task [Tagamets & Horwitz, 2001].

In new work, we explored the hypothesis that we can create an intermediate-scale neurocognitive system, but now one that includes regions from both hemispheres, working memory, and learning of interregional functional connectivity based on human fMRI data. Unlike the original functional imaging model, learning is now used heavily to acquire connection weights and pathway level inter-regional connectivity strengths instead of manually assigning such values. Most importantly, the new resulting model differs from most previous visual system models in being constrained to match quantitative fMRI data (some of which we collected ourselves), in spanning two cerebral hemispheres, and by its integration with hippocampal regions and with prefrontal working memory regions. To our knowledge, no one has previously developed a neurobiologically grounded computational model of delayed match-to-sample human behavior having this scope and fidelity to behavioral, neural, and functional imaging data.

Figure 7 depicts the overall architecture of the new extended model. The task that we model, visual shape matching, involves mainly the occipitotemporal visual pathway. Single-cell recordings in primates have provided data about specific visual response properties in these areas [Tanaka 1993], as have imaging studies in humans [Sergent et al 1992; McIntosh& Gonzalez-Lima 1994]. This pathway includes areas V1, V2, V4, the TEO region of the inferotemporal cortex (TEO/IT), and lateral prefrontal cortex (PFC). The hippocampus (HC) is primarily associated with long-term memory (LTM) but is also involved in working memory.

Each region in the model is composed of 8x12 arrays that represent subpopulations of neurons with different types of response properties. The early visual cortices (V1, V2) encode simple components of visual objects, such as line segments and their orientations. In the model, there are subpopulations that encode horizontal and vertical lines. Area V4 is the first region in the pathway that is considered to be association cortex, in which visual representations of basic shapes combine with other information. Area TEO/IT is thought to be a region of the brain that

encodes whole objects, such as faces, trees, or words, with specialization for different types of objects in different populations and sub-areas of TEO/IT. The PFC has been implicated in executive function in general, and is thought to contain abstracted representations of objects and their context. Finally, the hippocampus has also been implicated in WM, thought its role in this is not clearly understood. Neurons in the regions V1, V2, and V4 are active only when a stimulus is in view. Neurons in regions further along in the pathway (including areas TEO/IT and PFC) have the capability of maintaining high levels of activity even when no item is currently in view. Thus these regions are likely to play a key role in WM function. However, one distinction between areas TEO/IT and PFC that has been observed in neurons from electrophysiological experiments in monkeys is that WM traces are maintained across intervening stimuli in PFC, whereas in TEO/IT an intervening stimulus replaces the current memory with a representation of the new stimulus [Miller et al, 1993]. This suggests that neuronal circuits in the PFC implement the property of resistance to interference in WM. In the model, the PFC contains the WM circuits, and feedback from PFC to TEO/IT enhances temporary memory maintenance.

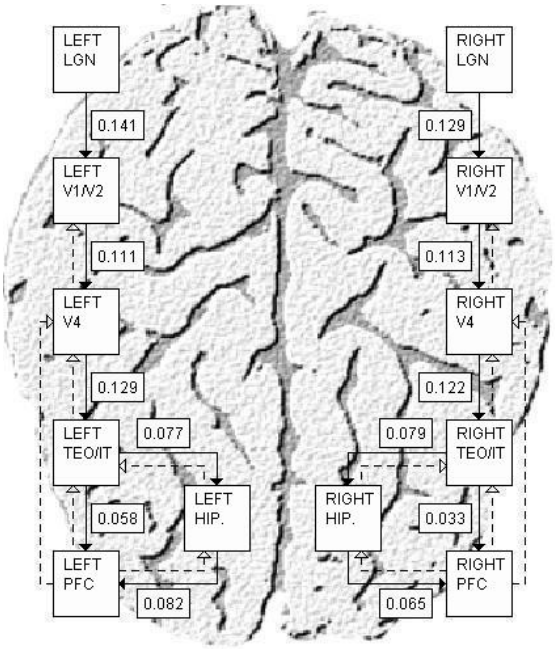


Figure 7: Architecture of the full model. Each block represents a brain region that has been implicated in visual working memory. Visual input enters the network through the lateral geniculate nucleus (LGN) and is passed forward through visual brain regions (V1, V2, and V4) that successively abstract the representations. Area TEO of the inferior temporal cortex (TEO/IT) is the region thought to be specialized for representations of whole visual objects. The prefrontal cortex (PFC) contains the working memory circuits, which maintain short-term memories of recent stimuli, and make decisions about whether they match the current stimulus.

Specialized circuits that maintain memory traces and decide on matching stimuli make up the populations in the PFC region of the model. The four different types of units in the WM circuit are based on distinct populations that have been identified in single-cell recordings in monkeys in delayed memory tasks [Funahashi & Kubota, 1994; Goldman-Rakic 1995]. A separate circuit implements decision-making in the model. This circuit is composed of two units in each hemisphere: one that responds when a stimulus matches the one in WM, and another that responds when there is no match. These units receive inputs from all of the WM circuit units in the frontal cortex, and thus collect the total response from all frontal WM circuits. The connection weights are determined by a supervised learning mechanism.

The most critical issue with the extended model was matching model activity to fMRI, which is an indirect measure of neuronal activity. The relationship between fMRI and neuronal measures is complex. The responses of neurons constitute the computations that are performed by the brain: a high firing rate in a neuron suggests selectivity of that neuron to a particular state of the brain, e.g., it might represent that a face is in view. The connections between neurons, i.e. their strengths and patterns, determine the responses of the neurons. Changing how neurons interact changes their firing properties and activity in these connections requires large amounts of energy. Energy requirements in local brain regions increase demand for oxygen. Finally, fMRI measures oxygen levels that change when blood flow responds to local changes in energy needs. The net effect is that fMRI is thought to mainly reflect the energy requirements of synaptic activity, and not the neuronal spiking that is commonly used as an index of encoding. We have previously demonstrated that the consequence of this is that there can be a dissociation between neuronal spiking activity and measured fMRI [Tagamets & Horwitz, 2001].

We developed a neural network learning algorithm, the *gains learning* algorithm, that can be used to find the strengths of interregional connections for the model to match activations in an arbitrary fMRI data set [Winder et al, 2007]. This problem differs from the usual supervised learning methods in neural networks in that there are target values for all regions in the network, not just for an output "layer." This method allows estimation of functional connectivity while allowing for effects of the interaction of interregional and local circuits. It differs from other measures of functional connectivity for fMRI currently in use in two major ways. First, the model itself is a generative one that attempts to explain how imaging data such as rCBF and BOLD can be explained by neuronal behaviors. Second, the connection training method is based on matching average activations in the regions of interest (ROIs), as opposed to other methods such as structural equation modeling [McIntosh & Gonzalez-Lima, 1994], partial least squares [McIntosh, 1998; McIntosh et al., 2004; McIntosh & Lobaugh, 2004], and others [Friston et al., 2003; Mechelli et al., 2003; Penny et al., 2004], which derive effective connection strengths from covariances or correlations that are computed from the data.

The gains learning algorithm is a gradient descent method that attempts to find solutions that will minimize the overall error between modeled and target activations. It was demonstrated empirically that this algorithm converges to unique solutions, and that it finds the correct solutions on data with known connectivity. We then applied it to an fMRI data set in order to examine connections in healthy control subjects as they performed a working memory task involving linguistic stimuli. The connection weights shown in Figure 7 provide an example. These specific weights were derived by using this learning method on the fMRI data. We also examined the effects of modifications in local prefrontal circuitry on fMRI activations,

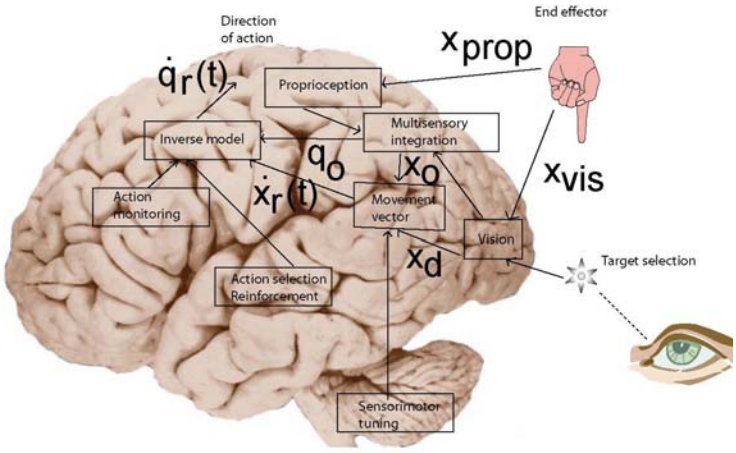
functional connectivity, and performance of the task. The results suggest that functional connections are much more sensitive to these changes than BOLD activations, and the performance changes are suggestive of working memory deficits commonly found in schizophrenia, in that the memory is more susceptible to interference. We conclude that our combined theory-driven and data-driven methodology extends current imaging analysis methods, and allows examination of properties other than total activations and functional interregional connection strengths that are currently in use for fMRI data analysis.

C. Adaptive Sensorimotor Control Model

A primary goal of research on the cognitive neuroscience of decision-making is to produce a comprehensive model of behavior that flows from perception to action (including decision-making) with all of the intermediate steps defined. The model should be able to generate not only simulated neural activity, fMRI and other functional neuroimaging data, but also behavioral performance (i.e., accuracy and reaction time data) data in both intact and neurological conditions. Although we and others (e.g., [Husain et al., 2004]) have developed models of perception, and models of action have also been put forward [Guigon and Baraduc, 2002; Contreras-Vidal and Wen, 2003], integrating perception, decision-making, and action networks is still needed. To address this gap, we have integrated a model of adaptive frontal-parietal sensorimotor transformation with redundant arm reaching. Our model now incorporates complementary parallel cortico-cerebellar-thalamo-cortical and cortico-striato-thalamo-cortical neural “loops” that are thought to be critical for motor adaptation learning in response to developmental and/or environmental changes.

The hypothesized cortical sensory integration and coordinate transformations required for controlling an arm reaching to visual targets (Figure 8) can be initially learned through simultaneous exposure to patterned proprioceptive and visual stimulation during self-produced movement [Guigon and Baraduc, 2002]. These sensorimotor transformations can then later be updated with the help of fronto-parietal and/or parieto-cerebellar circuits. Recent motor control theories suggest that the brain uses internal models to learn these mappings, and to plan and control accurate movements. An internal model is thought to represent how the biomechanics of the arm interacting with the outside world would respond to a motor command; therefore it can be seen as a predictive model of the reafference that helps the system plan ahead [Imamizu et al., 2000]. For example, during adaptation to external forces applied through a robotic manipulandum, these adaptive internal models are thought to generate compensating torques which allow the arm to track an invariant reference trajectory to a target. In the case of a distorted kinematic environment (e.g., altered screen cursor-hand relationships), the internal model would represent the new inverse kinematics required to transform a desired movement vector in visuospatial coordinates into a joint-based motor command. Adaptive sensorimotor behavior therefore involves the problems of localizing the hand and the targets in space, trajectory planning, coordinate transformation, and control, and the brain must solve these problems to bring the hand to a desired target location. A benchmark test performed by many researchers in motor learning is a reaching task between points usually lying along the circumference of a circle at equally spaced intervals. The experimenter either distorts the kinematic mapping of the handle (or mouse) or programs a robot handle to exert force disturbances on the subject. This allows researchers to examine how subjects react to various kinematic and dynamic perturbations.

Figure 8: Visual and proprioceptive signals (x_{vis}, x_{prop}) are integrated (x_o) and compared to the target location (x_d) to compute the movement vector. Next, computations in a parieto-premotor network transform changes in end-effector position into changes in joint angles specifying action direction. Deviations between desired and actual movements cause progressive recruitment of basal ganglia and cerebellar networks, updating the sensorimotor transformation networks.



An interesting result of these studies has been that despite large differences in models, they often display three common features: a trajectory generator, sensory feedback and control loops, and an adaptive process. However, past models cannot account for the effects of neurological lesions (e.g., Parkinsonism or cerebellar lesions) nor the functional changes resulting from environmental changes such as screen cursor rotations or force field perturbations.

To address these gaps, we implemented a model of redundant reaching by complementing the cortical circuit with two sub-cortical networks, namely, a fronto-striatal loop and a parieto-cerebellar loop. The fronto-striatal network is modeled as an adaptive search element, guessing new sensorimotor transformations and reinforcing successful guesses while punishing unsuccessful ones [Grosse-Wentrup and Contreras-Vidal, 2006]. This system uses an error (evaluative) signal to drive the selection and the reinforcement/punishment mechanisms. The parieto-cerebellar component is modeled as an adaptive error-correcting module that continuously updates a correction term to drive the error of actual versus desired movement to zero [Contreras-Vidal, Grossberg, Bullock, 1997]. Simulations of a kinematic visuomotor adaptation task using a redundant arm moving in the horizontal plane and with learning processes disabled in the cortico-striatal network resulted in error curves resembling those of Parkinson's disease patients [Contreras-Vidal and Buch, 2003; Grosse-Wentrup and Contreras-Vidal, 2006]. Simulated PET data have also been computed to assess the functional activation of various brain regions of interest [Contreras-Vidal and Wen, 2003]. The model's patterns of simulated constant and variable errors were found to match the learning curves seen in the experimental data. In agreement with experimental studies, for example, the simulated PET signal of superior parietal lobe showed an increase in functional activation due to the introduction of the visual feedback distortion.

D. Mini-RoboScout

We implemented a prototype agent, referred to here as "mini-RoboScout", to assess the feasibility of our neuromorphic framework outlined, to examine the issues that arise in combining different computational mechanisms, to further evaluate the use of a developmental approach, and to assess the adequacy of the three-tiered architecture described above. The central goal of this pilot study was to determine the implications of and barriers to this approach to creating an intelligent agent that is based upon integrating a variety of behavioral functions and computational mechanisms within a single framework. Our intent was to create a "skeletal system" that includes many of the needed components, focusing on the key issue of integrating these components in a coherent fashion, but that does not incorporate components that are as powerful as would be needed in a real environment. We kept the environment and input/output to the system relatively simple so that we could focus on the primary issue of integrating those components and not the important but low-level details that will eventually need to be addressed. Thus, for example, language input to our prototype agent consists of a sequence of phonemes rather than of an unprocessed acoustic signal, and the visual input has the various objects scattered around the environment color-coded to make their identification and separation from the environment much easier. Further, some aspects of learning are done off-line as a practical step. Ultimately more powerful auditory and visual processing methods and more online learning methods will replace those currently used. However, as long as we use a modular architecture as planned, such improvements should be viable and allow for the incremental and progressive enhancement of the system's performance. In summary, mini-RoboScout is a partial, exploratory implementation of the ultimate target system.

The basic scenario envisioned for mini-RoboScout is that of a grounded, embodied agent that interacts with simulated environments. Thus our pilot study involved implementation of both a practice environment as well as the prototype agent. At each discrete step of simulated time, mini-RoboScout receives an input image indicating what it can see from its current location, and also perhaps a sequence of auditory phonemes representing a heard spoken sentence. It then generates a movement and possibly a sequence of motor phonemes representing its spoken output. The state of the environment is then updated and the cycle begins again. Figure 9 shows an aerial view of the simulated setting, with a central "city" composed of roads and buildings. The agent does not have access to this map. Various objects are scattered around the environment, such as people and vehicles.

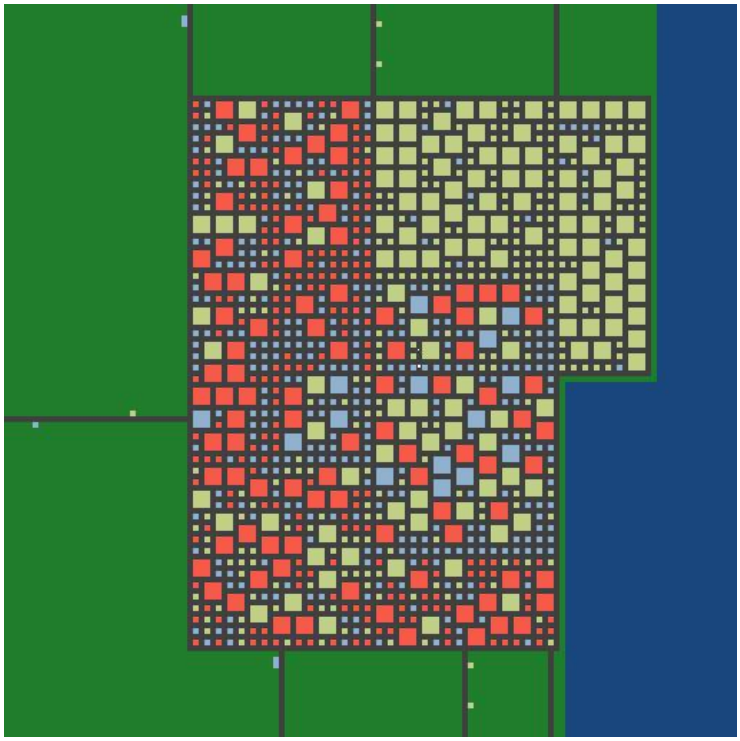


Figure 9: An "aerial view" of the simulated environment that represents an urban area in the central regions consisting of buildings (multicolored) and roads (black). Surrounding regions represent open fields (green) with scattered roads and small buildings, and a body of water (blue) on the right.

At each time step, the agent (mini-RoboScout) receives an updated image representing what it sees at its current location with its current orientation. The environment generates this image based on its current state (i.e., the agent's location and orientation, the map of the environment, and a database of existing objects). Figure 10 shows a single snapshot of the agent's current input image. The image is a 512 x 512 pixel array where each array entry is an RGB coded triplet representing the pixel's color. In this example the agent is in the urban area looking down a road. A stylized person (left near the front) and vehicle (down the road on the right) are represented by distinctive icons. As indicated earlier, these objects are currently color-coded to facilitate their identification so that the agent does not need to deal with the difficult issue of separating objects of interest from background. If the agent elects to focus on one of the agents in the environment, the environment automatically produces a corresponding sketch of the object that is used as input to the agent's visual system.

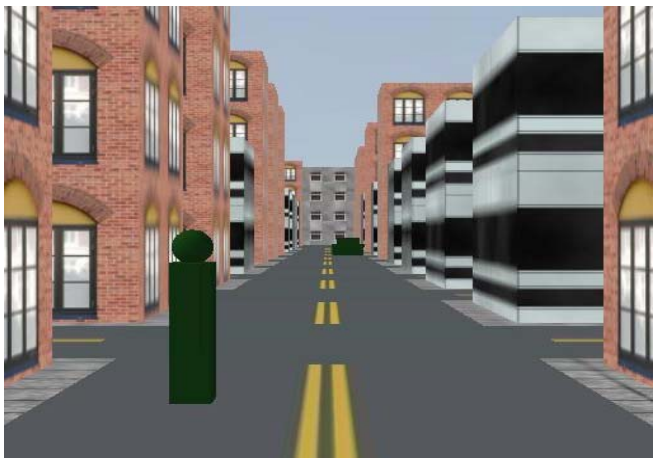


Figure 10: A snapshot of what the agent sees at a single step of a simulation. The agent is currently looking down a street somewhere in the urban region. This image contains two idealized objects, a person (left near the front) and a vehicle (further down the right side of the road) that are indicated by dark green icons. Visual input to the agent consists of a temporal sequence of such images, each determined by its position and location.

The other input to the agent during a time step is a sequence of auditory phonemes representing a spoken sentence that it hears. Each phoneme is encoded as a set of distinctive features, just as with the WLG model described above. On many time steps, no auditory input is received. Using the given visual and auditory input at a time step, the agent must *learn* to identify the objects present, interpret the spoken utterance, adjust its goals, determine its next incremental movement, and generate any appropriate spoken utterance.

The agent's controlling software is implemented as a three-tiered system as discussed earlier. The sensorimotor level consists of mostly neurocomputational components. After learning, the sensorimotor level in isolation (i.e., in the absence of a cognitive or executive level), when set in the state *MoveForward*, is capable of wandering the city, autonomously avoiding barriers to movement and noting objects that it encounters. This level of the system includes swarm intelligence methods for guiding movement control that we have found effective in past multi-agent systems [Winder 2004]. The cognitive level is where learning to classify object images and their significance is analyzed. Also, after learning, any input temporal sequence of phonemes undergoes a mapping process into its meaning. At present these auditory phoneme sequences consist primarily of simple commands, e.g., the phoneme sequence equivalent of "Go to the market district", where the prototype agent has been given the location of selected regions of the city a priori as rough coordinate boundaries. This level also generates spoken sentences when appropriate, and exerts top-down influences on the sensorimotor level's movement control to guide the agent to appropriate target locations. Finally, the executive level consists of two functions implemented as symbol-processing modules. A command memory stores recently received commands. The second component, a production system of control rules, examines the current situation and generates and prioritizes the prototype agent's goals.

Both the environment simulator and mini-Roboscout were implemented and tested [Winder, 2007]. If the agent kept either a local or cumulative memory of its nearby environment, its performance improved, both in navigating near obstacles and, with cumulative memory, when traversing previously visited city regions. These advantages persisted when a team of agents were used in the context of a pursuit scenario.

E. GPU Cluster Experiment

In an attempt to explore the computational power of GPUs, we developed a program written in C and Cg to simulate training a feed-forward neural network with a single hidden layer using error back-propagation. As noted earlier, performing computation on a GPU is challenging because operations must be mapped onto vertex and pixel shading operations. In order to pass information to the GPU, textures are created in which the relevant data is stored. Vertex and fragment programs are defined such that, when rendering takes place and the programs are executed, the desired computation is performed.

Consider, for example, calculating the weighted sum of I inputs to a hidden layer of size H, for all training examples from a training set of size T. To perform this computation using a GPU, two textures were defined. One texture was created that contained all of the input data for every training example. The dimension of this training-set-texture was T x I. A second texture was created to store the connection weights from every input neuron to every hidden neuron. This connection-weight-texture was of dimensions H x I. In order to compute the weighted sum of all of the inputs to the hidden layer, a rectangle with a width of H and a height of T was rendered. Texture coordinates were assigned to the rectangle, such that the lower left corner had texture coordinates of (0,0) while the upper right corner of the rectangle had texture coordinates of (H,T). After rendering was performed, a pixel in the i^{th} column and j^{th} row of this rectangle contained the weighted sum of the inputs to the i^{th} hidden neuron, for the j^{th} training set.

When rendering takes place, the fragment program being used is executed once for every pixel in screen space. An orthogonal projection and an appropriately sized viewport were used in order to ensure a one-to-one mapping between screen coordinates, texture coordinates, and geometry coordinates. A fragment program was written that received the two textures and the texture coordinates as parameters. Each time the fragment program was executed, it calculated the weighted sum of inputs for a particular hidden neuron for a particular training example. The fragment program was able to identify which hidden neuron and training set it was to perform the computation for by the texture coordinates that it received as parameters, and thus pulled the relevant information from the training-set-texture and connection-weight-texture. The fragment program calculated the weighted sum and used the resulting value to set the red value of the rendered pixel. Similar strategies were used to calculate the sigmoid function values, to calculate values for successive layers of the network, and to perform error back-propagation. Limitations on the length of the fragment program made it necessary to implement a multiple-pass rendering approach. For example, for a neural network with 128 input neurons, the weighted sum of all 128 input neurons could not be calculated in one execution of a fragment program. Therefore, an approach was taken wherein the weighted sum of the first 64 inputs was calculated, and then in a second rendering pass, the next 64 weighted sums were calculated and added to the previous result. This allowed us to perform the computation for networks of arbitrary size.

Rather than rendering to the screen, our implementation used an extension to OpenGL called framebuffer objects (FBO) to render to off-screen buffers. The use of FBOs is critical as it improves performance for a number of reasons. FBOs offer 32 bits of precision for floating point numbers, which is considerably higher than the 8 bits of precision offered by rendering to the screen. Additionally, using FBOs allows for rendering directly to a texture. This is crucial, as it means that throughout the computation process data may remain on the GPU and does not

need to be passed between the CPU and GPU, which is a very inefficient and slow process.

In order to evaluate the speed of our GPU implementation, a CPU-only version of a neural network, written in C, was also developed as a control. We applied these two different implementations to an image classification problem in order to compare performance. The training set for this problem consisted of 13 sketch images from an urban warfare setting. For each of these 64 by 64 pixel images, there were 13 possible observations to be made. Each image had a distinct combination of which observations were to be made.

A neural network with 4096 input neurons, 64 hidden neurons, and 13 output neurons was used to solve this classification problem. The GPU and CPU-only implementations were trained for 600 iterations. During each of these iterations, all training sets were evaluated and error back-propagation was performed. For each implementation, two such runs were performed. During each run the error of the networks was recorded, while in the other run the time it took to perform the iterations was recorded.

Error was measured for the two simulated networks over 600 iterations. The errors of the two networks are almost identical. This is important as it shows that the two identical networks are being simulated almost exactly the same on the two different implementations. Small discrepancies in network error late in the simulations were thought to be due to the accumulative effect of small differences in rounding error between the two implementations.

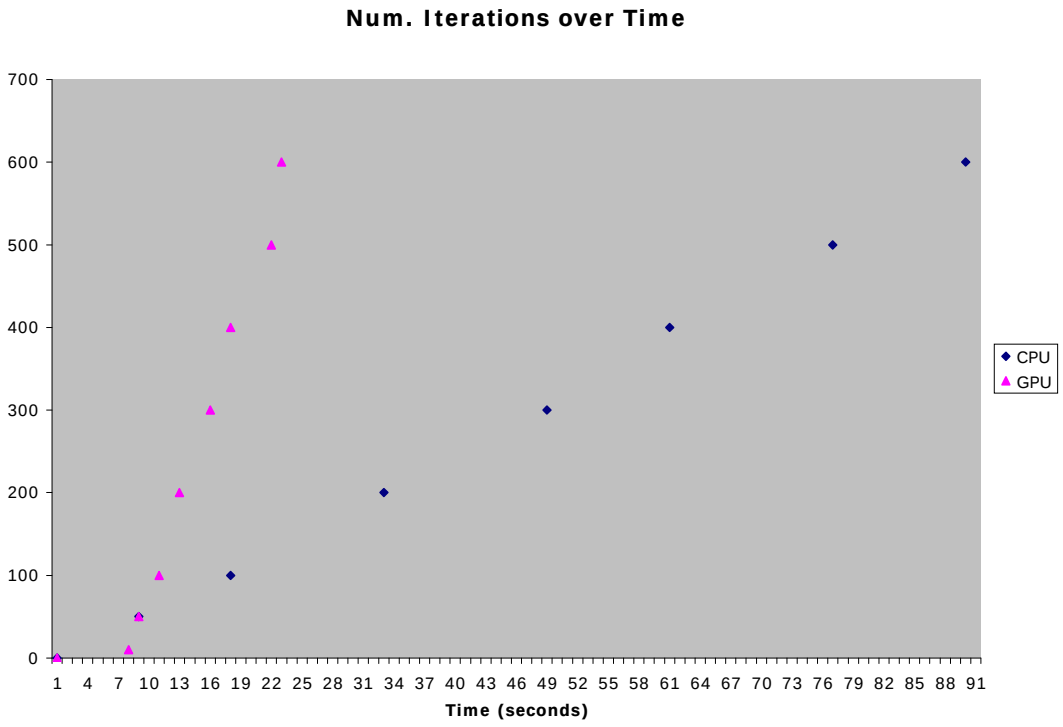


Figure 11: Iterations performed over time by GPU and CPU versions of neural networks.

Figure 11 displays how much time was required by each implementation to perform the 600 iterations. The GPU implementation started fairly slowly, likely due to one-time start up costs such as creating and binding textures, but quickly surpassed the CPU implementation. In the first 8 seconds, the GPU implementation performed the same number of iterations as the CPU implementation. As more time passed, the GPU implementation significantly outperformed the CPU implementation. While it took the CPU implementation 89 seconds to perform 600 iterations, the GPU implementation took only 22 seconds.

While promising, these results only reflect the performance of the two implementations on one particular set of training data and when simulating one particular network. Additional experiments were done to evaluate the performance of the two implementations on varying random training data. In each of the experiments, the size of the input layer was fixed at 4096, the size of the output layer was fixed at 32, and the size of the sets of training examples (randomly generated) was fixed at 32. The size of the hidden layer was varied between 64, 128, 256, 512, and 1024. Table 1 shows the performance of the two implementations during these simulations. The GPU outperformed the CPU implementation in every trial. Further, the difference in performance appears to scale in proportion to the size of the hidden layer.

Table 1: Time to Perform 100 Iterations by CPU and GPU Implementations

Hidden Layer Size	CPU Time (seconds)	GPU Time (seconds)
64	51	10
128	120	13
256	213	21
512	452	35
1024	967	62

On average, the GPU implementation simulated the neural network 10.6 times faster than the CPU implementation. This is below the improvement by a factor of 20 that was reported by [Oh and Jung 04]. However, there are some improvements that may be made to the current implementation that have the potential to drastically improve performance. The current implementation fails to take advantage of the data types supported by the GPU's specialized hardware. Specifically, using all four channels of textures and the float4 data type, which packs four floating-point numbers into one data member and may be used to perform dot and cross products very quickly, offers the potential for drastic speed improvements when computing weighted sums. These improvements have the potential to push the performance of our implementation well past the speedup by a factor of 20.

F. Evolution of Recurrent Networks

We undertook a pilot study using genetic programming (GP) as a design tool to assist with creating a recurrent neural network for sequence processing. Our goal was to critically assess the potential of this approach to help optimize the components used in a large scale neurocognitive architecture. In performing this "computational experiment", we used a general purpose software environment for evolving neural network architectures that we developed at the University of Maryland [Jung & Reggia, 2006]. This system emphasizes the integration of evolutionary processes with developmental and learning processes, and it supports the creation of modular

networks. To use this system, one specifies a class of neural network architectures that are to be considered. In other words, the space of all neural networks is too large to search, so one instead indicates the class of architectures to be considered by the evolutionary process. This is done using a high-level descriptive language to indicate the sets of modules (layers), allowable inter-module pathways, and other aspects of a neural architecture that may potentially be included as part of an architecture. Following an initialization step in which a random population of genotypes is created within the search space, the evolutionary process then involves a repeated cycle of three stages: development, learning, and genetic operations. The development stage literally grows each neural network (phenotype) from its high-level description; the learning stage then lets weight changes occur during a learning process based on data relevant to the specific task at hand; and finally, the fitness of each network is assessed followed by fitness-guided non-deterministic selection of parents from the environment and mutation, crossover and reproduction (producing the next generation). Fitness criteria may reflect not only network performance on the task at hand (e.g., mean squared error in pattern classification), but also measures of the network's properties (e.g., total number of nodes/connections).

The specific task we considered in this context is the problem of learning a temporal sequence of phoneme outputs that correspond to a given fixed input word pattern. For example, given the word *apple* as a fixed input pattern of five written letters A P P L E, the neural network should learn to generate the phoneme sequence /ae/, /p/, and /l/ followed by an end-of-word break in the output. A set of 230 variable length (2 to 6 phonemes) words were use for training. The space of neural networks to be considered by the evolutionary process is as follows. The fixed part of the structures is a feed-forward, three-layer network consisting of input, hidden and output layers as depicted on the right side of the illustration. The parts of the architecture to be evolved included additional hidden/delay layers and their connectivity. Feedback comes from core hidden and output layers, but where that feedback goes, how many hidden delay layers are used, etc. in the feedback process are evolved. Fitness of networks was based on two cost measures: root mean squared error (RMSE) to assess performance, and the total sum of absolute values of network weights following learning to penalize larger networks. A multi-objective evolutionary algorithm (SPEA) was used.

We ran a total of 100 evolutionary processes, each time having a population size of 25 networks and involving 50 generations. The initial population of neural network architectures and their initial weights were determined randomly (and thus differed) in each run. Only mutation was used as a genetic operator. The results are shown in Figure 12, averaged over the same architectures (i.e., each point in the figure represents a network having a specific number of hidden and output delays, and a specific layer-to-layer connectivity). The two fitness criteria are on the axes: network RMSE on the vertical axis, and sum of network weight magnitudes on the horizontal axis. Following the Pareto optimal front (solid line) downwards from the upper left, one has initially very simple networks with only one or two hidden delay layers. As networks get additional hidden delay layers and connectivity, one gets better performance. In other words, moving rightwards along the solid line gives progressively more accurate, but more complex networks. Figure 13 shows two examples of networks evolved in this fashion. The network labeled Ho_2 has two evolved delay layers receiving feedback from its fixed hidden layer on the right, with both sending their activity to the output layer directly. The network labeled Hh_1Oo_1 has two evolved delay hidden layers sending/receiving connections from the fixed hidden and

output layers in network's center. The point is that even though these two networks have the same numbers of hidden delay layers and the same number of pathways (arrows), they have markedly difference performance measurements, with the network Hh_1Oo_1 being able to learn to generate correct phoneme sequences qualitatively better than the network Ho_2 can do. These and other similar insights were not at all apparent prior to doing the evolutionary runs, and to our knowledge have not been demonstrated in past neural network studies.

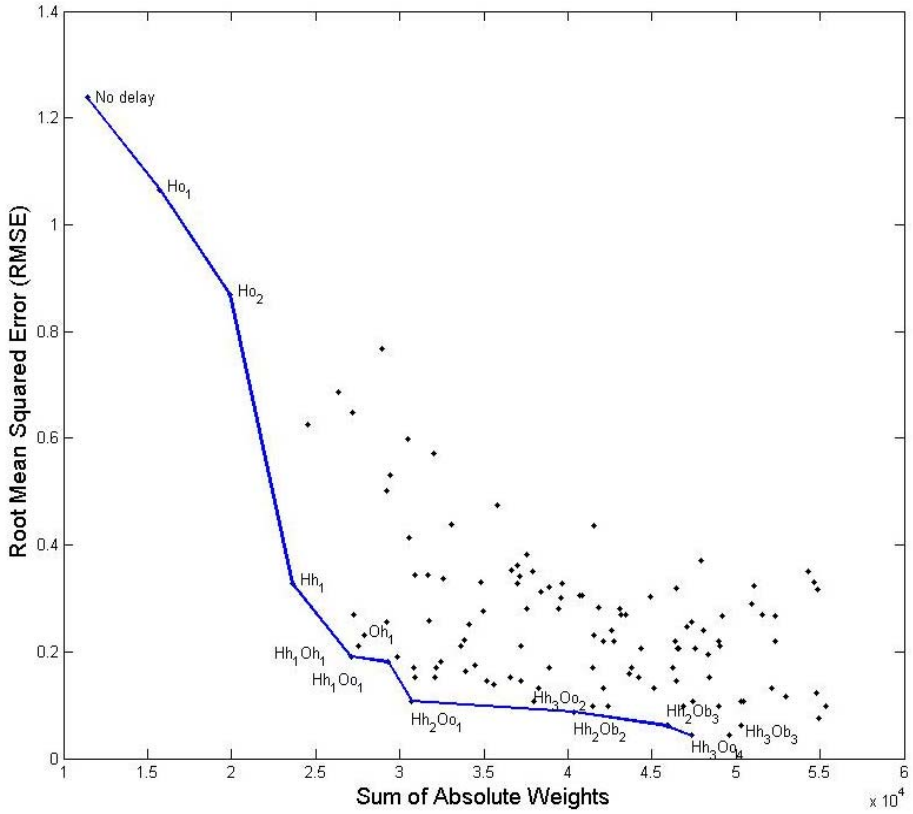


Figure 12: The results for networks from all final generations of 100 runs of the evolutionary process are shown. Each plotted point represents one network architecture's values averaged over all evolutionary runs. Points on the solid line represent the Pareto optimal set.

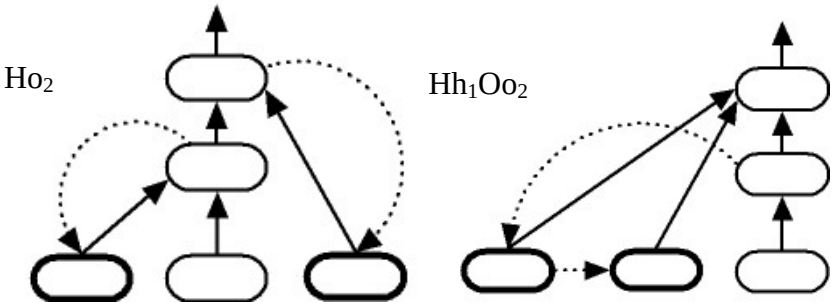


Figure 13: Example evolved architectures. The evolved layers are shown as bold ovals. Dotted lines are feedback connections, solid arrows are trainable fully-connected paths.

The results from this study demonstrate the ability of evolutionary processes based on GP to discover parsimonious but effective network architectures for specific given tasks. We conclude from this exercise that GP may have a significant role to play in designing or optimizing components of a cognitive architecture based upon neuromorphic principles.

G. Roadmap

How should the development of a large-scale, integrated neurocognitive architecture be organized over a several year period? Our answer is that there should be three aspects to implementation and evaluation of such a system. First, a *full skeletal system* should be implemented. By this we mean that all of the components needed for the full architecture should be in place ("full"), but that none of these components will necessarily be optimal. This is a somewhat non-standard approach that can be viewed as an extension and completion of the prototype "mini-Roboscout" system described earlier. The philosophy underlying this approach is that the integration of the components of the core architecture needs to be achieved first as it is the critical step upon which ultimate success will depend.

Once this full skeletal system is functioning effectively, the second aspect of the implementation process would be the gradual replacement of the initial components of the architecture with progressively more powerful components, using the best technological solutions available in each case. The need to be able to swap in improved components like this is one of the reasons for requiring a highly modular design. There is a natural ordering to this component enhancement that progresses from sensorimotor to cognitive to executive functions.

Finally, the third aspect of the implementation process is a concurrent *research process* that addresses fundamental research issues that will be faced in producing a full, integrated architecture. A key example from our perspective is how those operations originally implemented using symbolic methods (e.g., hierarchical partial planner at the executive level) can progressively be replaced by neurocomputational mechanisms). Another issue to be addressed is the optimization of selected components/subsystems using genetic programming methods. The results from these concurrent research efforts will directly feed into and influence the full system implementation.

Discussion and Conclusions

We have outlined both a conceptual framework and a top-level design for an integrated cognitive architecture, and tested several of the ideas that are involved through some computational experiments. Our principal conclusions from this work are as follows.

1. A large-scale, integrated neurocognitive architecture is feasible. By this we mean that knowledge in neuroscience and advances in computational power make it plausible that a general purpose machine intelligence can be developed that is directly based on the hierarchical and modular organization, dynamics and plasticity of the human brain. We outlined the different brain modules and functionality that need to be captured in such an architecture.

2. While such a neuromorphic architecture is the ultimate target, our currently incomplete knowledge about the neurobiological basis of cognition suggest that the optimal approach for the short term of the next few years should focus on a hybrid system that combines both neurocomputational and other "bottom-up" methods with symbolic and other "top-down" methods from AI and cognitive science. Such a hybrid approach is most likely to be reasonably successful when assessed critically by performance evaluations, and would be a natural spring-board for a long term effort over decades to produce a fully neuromorphic system.

3. Concurrently with development of a hybrid cognitive architecture for the short term, basic research efforts should be made to precisely define and to remove the remaining barriers to implementing a fully neuromorphic system.

4. The organization of the human nervous system suggests that a three-tiered system like that we have specified, consisting of sensorimotor, cognitive and executive levels, is a very useful approach to organizing implementation of the hybrid architecture. Other key biologically-inspired concepts include the use of developmental principles to guide the staged creation of the system and the use of a highly modular design.

5. Two non-standard computational ideas are likely to make substantial contributions to implementation of a neurocognitive architecture. First, over the short term, the use of a coarse-grained, high-performance computer cluster is probably the most cost effective approach to providing the needed computing power. Second, the use of evolutionary computation methods such as genetic programming as a design tool is likely to suggest efficient and novel neural network designs for use in the cognitive architecture. Nanotechnology and DNA computing offer additional promise for the long term.

Of particular interest are the intermediate-scale models that we studied. Training these models was computationally tractable: learning times were measured in hours using contemporary computers. The primary conclusion from these results is that one can readily build substantial portions of system-level models of basic aspects of cognition at present using the framework and principles that we propose. More specifically, the results that we obtained establish several important aspects of our conceptual framework by showing the following. First, it is possible today to routinely assemble networks of regions whose functionality is not pre-assigned or programmed-in, but is determined during learning by their location within a network of interconnected regions. Second, temporal sequences can be recognized and generated appropriately by such networks following training, based on recurrent connectivity between

regions. Third, a learning agenda can be used to divide the learning process into manageable pieces, allowing an entire system to learn in a multi-step process that resembles the occurrence of multiple stages during human childhood development. Fourth, working memory can readily be implemented as regional activation attractor states, activation patterns that persist across multiple input/output events. Fifth, learning of higher-level pathway weights (gains) can be guided effectively via data about functional connectivity of brain areas collected during experimental fMRI studies. Sixth, the complexity of these systems makes it very difficult to monitor their dynamics and changes during learning; a graphic interface permitting visualization of model states is very informative and increasingly necessary as the size of a system increases.

Literature Cited

Abbott L & Regehr W. Synaptic Computation, *Nature*, 431, 2004, 796-803.

Anderson J et al, Integrated Theory of Mind, *Psych. Rev*, 111, 2004, 1036-60.

Anderson, J., R. Gilmore, et al. (1999). Conduction aphasia and the arcuate fasciculus: A reexamination of the Wernicke-Geschwind model. *Brain and Language* 70(1): 1-12.

Banzhaf W, Nordin P, Keller R & Francone F. *Genetic Programming*, Morgan Kaufmann, 1998.

Bernhard F and R. Keriven. Spiking Neurons on GPUs. Research Report 05-15, CERTIS-ENPC, Ecole Nationale des Ponts et Chaussees, 2005.

Bi G and Poo M. Synaptic Modification by Correlated Activity, *Annual Review of Neuroscience*, 24, 2001, 139-166.

Binder, J., J. Frost, et al. (1997). Human brain language areas identified by functional magnetic resonance imaging. *Journal of Neuroscience* 17(1): 353-362.

Booth, J., D. Burman, et al. (2002). Functional anatomy of intra- and cross-modal lexical tasks. *NeuroImage* 16(1): 7-22.

Brachman R, Levesque H. *Knowledge Represent. & Reasoning*, Morgan-Kaufmann, 2004.

BrookGPU is available at: http://graphics.stanford.edu/projects/brookgpu/index.html

Brown C & Hagoort P. *Neurocognition of Language*, Oxford Univ. Press, 1999.

Caplan, D. (2003). Aphasic syndromes. *Clinical Neuropsychology*. K. Heilman and E. Valenstein. New York, Oxford University Press**:** 14-34.

Contreras-Vidal JL & Gold DR. (2004) Dynamic estimation of hand position is abnormal in Parkinson's disease. *Parkinsonism and Related Disorders,* 10(8):501-506.

Contreras-Vidal JL & Kerick S. (2004). Independent component analysis of dynamic brain responses during visuomotor adaptation. *Neuroimage*. 21(3): 936-945

Contreras-Vidal JL & Schultz S. (1999). A predictive reinforcement model of dopamine neurons for learning approach behavior. *J Comput Neurosci*. 6(3):191-214.

Contreras-Vidal JL, Grossberg S, Bullock D (1997) A neural model of cerebellar learning for arm movement control. *Learning and Memory*, 3(6):475-502.

Contreras-Vidal J & Buch E (2003). 'Effects of Parkinson's disease on visuo-motor adaptation'. *Exp Brain Res* 150:25–32.

Contreras-Vidal J & Wen J (2003). Functional Activation, Proc Intl Graph. Soc., 72-76.

DeJong K. *Evolutionary Computation*, MIT Press, 2006.

Dronkers, N., D. Wilkins, et al. (2004). Lesion analysis of the brain areas involved in language comprehension. *Cognition* 92: 145-177.

Friston, K. J., Harrison, L., & Penny, W. (2003). Dynamic causal modelling. *NeuroImage*, 19, 1273-1302.

Funahashi, S. & Kubota, K. (1994). Working memory and prefrontal cortex. *Neurosci.Res.*, 21, 1-11.

Ghallib M, Nau D & Traverso P. *Automated Planning*, Morgan-Kaufmann, 2004.

Gibbons A. The Brain's Energy Crisis, *Science*, 280, 1998, 1345-7.

Goldman-Rakic, P. S. (1995). Cellular basis of working memory. *Neuron*, 14, 477-485.

Grosse Wentrup M & Contreras-Vidal JL. (2006). The role of the striatum in adaptation learning: A computational Model. Submitted to *Biological Cybernetics*.

Gruau F, Whitley D , Pyeatt L. A Comparison Between Cellular Encoding and Direct Encoding

for Genetic Neural Networks, in *Proc First Ann. Conf. Genetic Programming, 1996, 81-9.*
Grundstrom E & Reggia J. Learning Activation Rules, *Int. J. Neural Sys*, 7, 1996, 129-47.
Guigon E & Baraduc P (2002). A neural model of perceptual-motor alignment. *J Cogn Neurosci* 14:538–549.
Husain F, Tagamets M, et al (2004) Relating neuronal dynamics for auditory object processing to neuroimaging activity: a computational and fMRI study. *NeuroImage*, 21**:** 1701-20.
Imamizu H, Miyauchi S, Tamada T, et al. (2000) Human cerebellar activity reflecting an acquired internal model of a new tool. *Nature* 403(6766):192-5.
Jan C, et al, A 65nm Ultra Low Power Logic Platform Technology using Uni-axial Strained Silicon Transistors, *Intnl. Electron Devices Meeting, Technical Digest*, Dec. 5, 2005, 60.
Jung J. & Reggia J. Evolutionary Design of Neural Network Architectures Using a Descriptive Encoding Language, *IEEE Trans. On Evolutionary Computation*, 2006, in press.
Kagan J & Baird A. Brain and Behavioral Development During Childhood, in *The Cognitive Neurosciences III*, M. Gazzaniga (ed.), MIT Press, 2004, 93-103.
Kagerer FA, Contreras-Vidal JL, Stelmach GE (1997) Adaptation to gradual as compared with sudden visuo-motor distortions. *Exp Brain Res* 115:557–561.
Kawato M, Wolpert D (1998) Internal models for motor control. *Novartis Found Symp* 218:291–304.
Krakauer JW, et al (2004) Differential cortical and subcortical activations in learning rotations and gains for reaching: A PET Study, *J of Neurophysiology*, 91:924-933.
Lohn J & Reggia J. Automatic Discovery of Self-Replicating Structures in Cellular Automata, *IEEE Trans. On Evolutionary Comp.*, 1, 1997, 165-178.
Long, L. and Gupta, A., Scalable Massively Parallel Artificial Neural Networks, *AIAA Paper* No. 2005-7168, AIAA InfoTech@Aeropace Conference, Sept., 2005, Wash., D.C.
Markram H, Luebke J, et al. Regulation of Synaptic Efficacy by Coincidence of Post-synaptic aps and epsps, *Science*, 275, 1997, 213-215.
McIntosh, A. R. (1998). Understanding neural interactions in learning and memory using functional neuroimaging. *Ann.N.Y.Acad.Sci.*, 855, 556-571.
McIntosh, A. R., Chau, W. K., & Protzner, A. B. (2004). Spatiotemporal analysis of event-related fMRI data using partial least squares. *NeuroImage*, 23, 764-775.
McIntosh, A. R. & Gonzalez-Lima, F. (1994). Structural equation modeling and its application to network analysis in functional brain imaging. *Human Brain Mapping*, 2, 2-22.
McIntosh, A. R. & Lobaugh, N. J. (2004). Partial least squares analysis of neuroimaging data: applications and advances. *NeuroImage*, 23 Suppl 1, S250-S263.
Mechelli, A., Price, C. J., Noppeney, U., Friston, K. J. (2003). A dynamic causal modeling study on category effects: bottom-up or top-down mediation? *J.Cogn Neurosci.*, **15**, 925-934.
Mesulam M, Large-Scale Neurocognitive Networks, *Ann Neurol*, 28, 1990, 597-613.
Middleton FA, Strick PL. (2000) Basal ganglia and cerebellar loops: motor and cognitive circuits. *Brain Res Brain Res Rev.* (2-3):236-50.
Miller, E. K., Li, L., & Desimone, R. (1993). Activity of neurons in anterior inferior temporal cortex during a short- term memory task. *Journal of Neuroscience*, 13, 1460-1478.
Mountcastle V. *The Cerebral Cortex*, Harvard Univ. Press, 1998.
Neville H & Bavelier D. Specificity and Plasticity in Human Neurocognitive Development, *The New Cognitive Neurosciences*, M. Gazzaniga (ed.), MIT Press, 2000, 83-98.
Nielsen M & Chuang I. *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

Oh K. and K. Jung. GPU Implementation of Neural Networks. *Pattern Recognition*, Vol. 37, No. 6, pp. 1311-1314, 2004.
Pan Z & Reggia J. Artificial Evolution of Arbitrary Self-Replicating Structures, *Journal of Cellular Automata*, 2, 2006, 105-123.
Passingham R, Stephan K & Kotter R. The Anatomical Basis of Functional Localization in the Cortex, *Nature Reviews Neuroscience*, 3, 2002, 606-616.
Penny, W (2004). Modeling functional integration: a comparison of structural equation and dynamic causal models. *NeuroImage*.
Pethick, M., et al. Parallelization of a backpropagation neural network on a cluster computer.
Proc. of 15th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2003). Marina Del Rey, California (2003) pp. 574-582
Poeppel, D. and G. Hickok (2004). Towards a new functional anatomy of language. *Cognition* **92**: 1-12.
Rao R & Sejnowski T. Predictive Learning of Temporal Sequences in Recurrent Neocortical Circuits. In Solla S et al (eds.), *Advances in Neural Information Processing Systems*, MIT Press, 12, 2000, 164-171.
Ravi S. Prasher et al., Nano and Micro-Technology-Based Next-generation Package-level Cooling Solutions, *Intel Technology Journal*, 9 (4), Nov. 9, 2005, 285.
Raz A & Buhle J. Typologies of Attentional Networks, *Nature Rev. Neurosci*, 7, 2006, 367-379.
Reggia J et al. Competitive Distribution in Neocortex, *Neural Comp.*, 4, 1992, 287-317.
Reggia J, Goodall S, & Shkuro Y. Computational studies of lateralization of phoneme sequence generation, *Neural Computation*, 10, 1998, 1277-1297.
Reggia, J., S. Goodall, et al. (2001). The callosal dilemma: Explaining diaschisis in the context of hemispheric rivalry via a neural network model. *Neurological Research* **23**: 465-471.
Riedmiller, M. H. Braun (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Proceedings of the IEEE Conference on Neural Networks.*
Rodriguez A. & Reggia J. Collective Movement Teams for Cooperative Problem Solving, *Integrated Computer-Aided Manufacturing*, 12, 2005, 217-235.
Rosenbloom P, Laird J & Newell A, *The Soar Papers*, MIT Press, 1993.
Russell S & Norvig P, *Artificial Intelligence*, Prentice Hall, 2003.
Schulz, R. and J. Reggia (2004). Temporally asymmetric learning supports sequence processing in multi-winner self-organizing maps. *Neural Computation* **16**(3): 535-561.
Seiffert, U. Artificial Neural Networks on Massively Parallel Computer Hardware. *European Symposium on Artificial Neural Networks*, pp. 319-330, 2002.
Sergent, J., Ohta, S., & Macdonald, B. (1992). Functional neuroanatomy of face and object processing: A positron emission tomography study. *Brain*, **115**, 15-36.
Shkuro, Y., M. Glezer, et al. (2000). Interhemispheric effects of simulated lesions in a neural model of single word reading. *Brain and Language* **72**: 343-374.
Shkuro Y & Reggia J. Cost During Evolution, *Cognitive Sys Res*, 4, 2003, 365-83.
Silva G. Neuroscience Nanotechnology, *Nature Reviews Neuroscience*, 7, 2006, 65-74.
Snodgrass J, Vanderwart M (1980) A Standardized Set of 260 Pictures: Norms for Name Agreement, Image Agreement, Familiarity, and Visual Complexity. *Journal of Experimental Psychology: Human Learning & Memory*, **6**, 174-215.
Sowa J. *Knowledge Representation*, Brooks/Cole, 2000.
Sutton R & Barto A. *Reinforcement Learning*, MIT Press, 1998.
Tagamets MA, Horwitz B (1998) Integrating electrophysiological and anatomical experimental

data to create a large-scale model that simulates a delayed match-to-sample human brain imaging study. *Cereb. Cortex*, 8**:** 310-320.

Tagamets M & Horwitz B. A model of working memory, *Neural Networks*, 13, 2000, 941-952.

Tagamets, M. A. & Horwitz, B. (2001). Interpreting PET and fMRI measures of functional neural activity: the effects of synaptic inhibition on cortical activation in human imaging studies. *Brain Res.Bull.*, **54**, 267-273.

Tanaka, K. (1993). Neuronal mechanisms of object recognition. *Science*, **262**, 685-688.

Tinnirella M, Tagamets M, Weems S, Contreras-Vidal J and Reggia J. *A Behavior-to-Brain Map*, CS-TR-4803/UMIACS-TR-2006-24, University of Maryland, 2006.

Turing A. Computing Machinery and Intelligence, *Mind*, 59, 1950, 433-460.

Tyagi S, Auth C, Bai P, et al, An Advanced Low Power, High Performance, Strained Channel 65nm Technology, *International Electron Devices Meeting, Tech. Digest*, Dec. 5 2005

Uttal W. *The New Phrenology*, MIT Press, 2001.

Vouloumanos, A. and J. Werker (2004). Tuned to the signal: The privileged status of speech for young infants. *Developmental Science* 7(3): 270-276.

Weems, S, J. Reggia (2004). Hemispheric specialization and independence for word recognition: A comparison of three computational models. *Brain and Language* 89: 554-568.

Weems S & Reggia J. Simulating single word processing in the classic aphasia syndromes based on the Wernicke-Lichtheim-Geschwind Theory, *Brain and Language*, 98, 291-309.

Winder R. The Influence of Collective Working Memory Strategies on Agent Teams, PhD Dissertation, Dept. of Computer Science, University of Maryland, August 2007.

Winder R, Cortes C, Reggia J & Tagamets M. Functional Connectivity in fMRI: A modeling Approach, 34, 2007, 1093-1107..

Winder R & Reggia J. Using Distributed Partial Memories to Improve Self-Organizing Collective Movements, *IEEE Transactions On Systems, Man and Cybernetics (B)*, 34, 2004, 1697-1707.

Yao X. Evolving Artificial Neural Networks, *Proc. IEEE*, 87, 1999, 1423-1447.

Zhu, J and Sutton, P. FPGA Implementation of Neural Networks - A Survey. In Cheung, P, Constantinides, G. and de Sousa, J., Eds. *13th International Conf. on Field-Programmable Logic and Applications*, 2003, 1062-1066.

List of Symbols, Abbreviations and Acronyms

AF	arcuate fasciculus
AG	angular gyrus
AI	artificial intelligence
A1	primary auditory cortex
BA	Broca's area
BOLD	blood oxygen level dependent (imaging)
CPU	central processing unit
EEG	electroencephalogram
FBO	frame buffer objects
fMRI	functional magnetic resonance imaging
GHZ	giga-Hertz
GP	genetic programming
GPU	graphical processing unit
IC	integrated circuit
IT	interior temporal region
MOSFET	metal oxide semiconductor field-effect transistor
M1	primary motor cortex
PFC	prefrontal cortex
rCBF	regional cerebral blood flow
RMSE	root mean square error
ROI	region of interest
RPROP	resilient error backpropagation
SMG	supra-marginal gyrus
S1	primary somatosensory cortex
TEO	inferior temporal cortex
V1	primary visual cortex
V2	secondary visual cortex
WA	Wernicke's area
WLG	Wernicke-Lichtheim-Geschwind
WM	working memory
1D	one dimensional
2D	two dimensional
3D	three dimensional